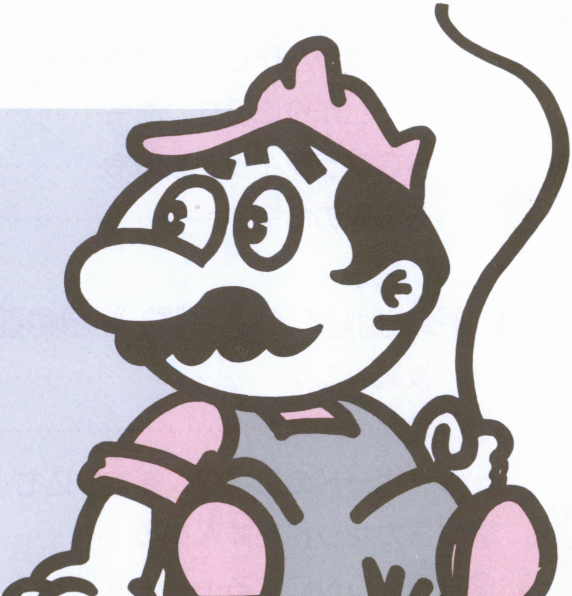


Nintendo®

ファミリーベーシック

FAMILY BASIC™ V3

ハンドブック



(S60年)
1985年3月6日(水)

ごあいさつ

このたびは，任天堂『ファミリーベーシックV3』カセットをお買いあげいただきまして、誠にありがとうございました。

ご使用の前に，取り扱い方，使用上の注意等，このハンドブックをよくお読みいただき,正しい使用方法でご愛用ください。

なお，このハンドブックは，大切に保管してください。

もくじ

1. FAMILY BASIC(ファミリーベーシック)™ V3の特徴(とくちょう)を知(し)ろう！



1-1 4つのプログラムを内蔵(ないぞう)しています

ファミリーベーシックV３のカセットには，ＢＡＳＩＣ(ベーシック)と一緒(いっしょ)に４つのプログラムが入(はい)っています。

①ハート

②ペンペン迷路(めいろ)

③マリオワールド

④スターキラー

これらのプログラムは，電源(でんげん)スイッチを入(い)れたあと，F１〜F４のキーを使(つか)って，ワンタッチで 実行(じっこう)することができます。IIコントローラーについているマイクに息(いき)を吹(ふ)き込(こ)むとハートのマークが出(で)てきたり，ペンペンやマリオ，宇宙船(うちゅうせん)などが登場(とうじょう)する楽(たの)しいものばかりです。カセットテープからプログラムを読(よ)み込(こ)む手間(てま)もかかりませんから，いつでも気軽(きがる)に遊(あそ)ぶことができます。

さらに，これらのプログラムはすべてＢＡＳＩＣ(ベーシック)で書(か)かれていますから，自由(じゆう)に改造(かいぞう)することができます。ふつうは，本(ほん)に載(の)っているプログラムを改造(かいぞう)するというのは，けっこう手間(てま)のかかるものなのですが，この本(ほん)ではそれぞれのプログラムについて，詳(くわ)しく改造(かいぞう)方法(ほうほう)を紹介(しょうかい)していますから，初(はじ)めての人(ひと)でも簡単(かんたん)に改造(かいぞう)を楽(たの)しむことができます。このような方法(ほうほう)ならば，最初(さいしょ)からプログラムを作(つく)ることなく，自分(じぶん)だけのオリジナルゲームを作(つく)ることができるわけです。

なお，内蔵されている4つのプログラムについては，全部のリストとその説明がしてあります。細かい点まで理解するのは難しいかも知れませんが，プログラムの全体的な流れはつかめると思います。ぜひ，BASICの勉強や自分でゲームを作るときの参考にしてください。

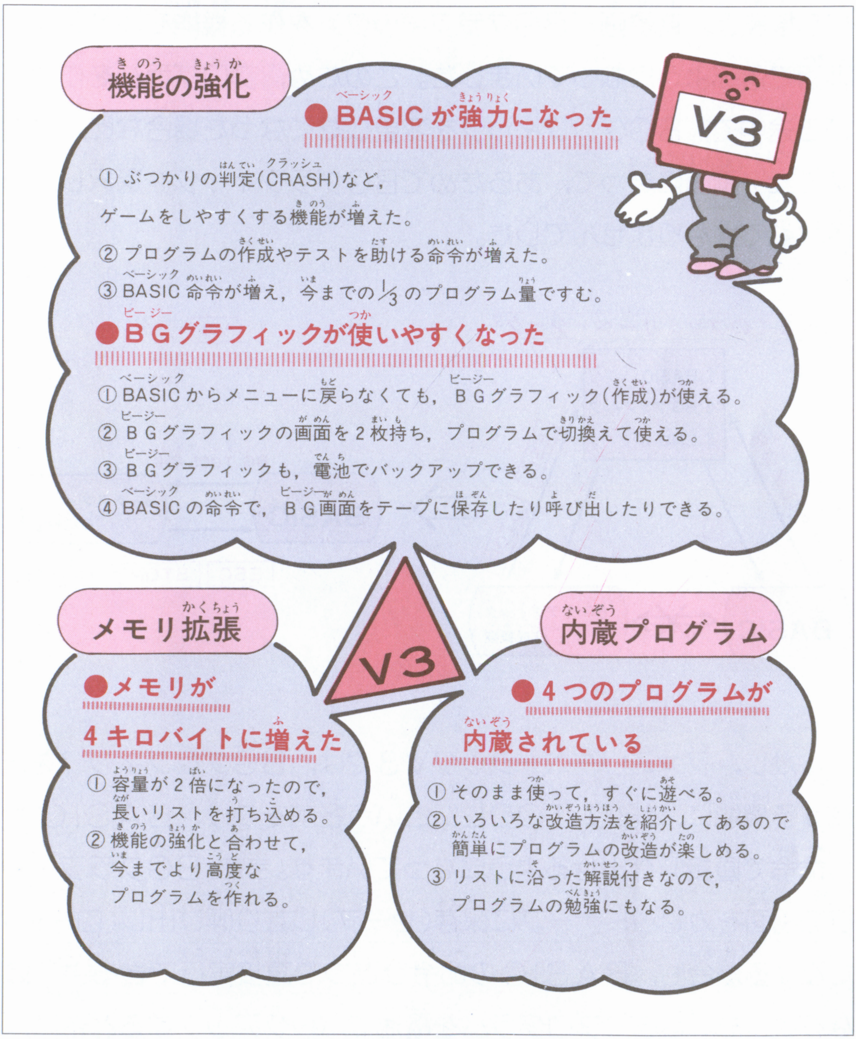


図1

1-2 BGグラフィックが使いやすくなりました

今までのファミリーベーシックでは，BASICのプログラムを作る機能と，背景画（BGグラフィック）を作る機能を，メニュー画面で選ぶようになっていました。このため，プログラムを作っている途中で，BGグラフィックを変更したくなった場合など，1度メニュー画面に戻って，あらためてBGグラフィックを選択しなおさなくてはなりませんでした。

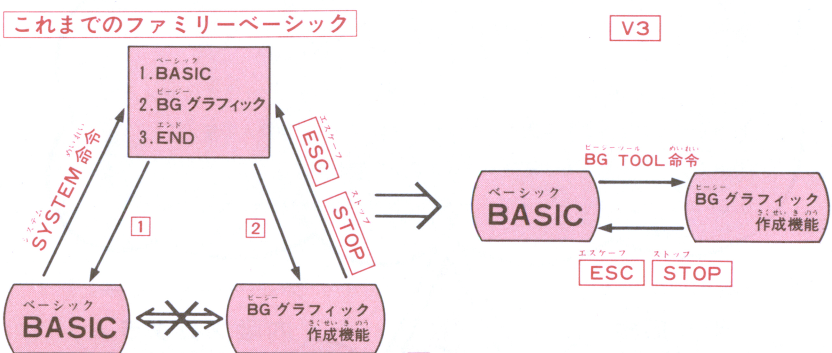


図 2

しかし，ファミリーベーシックV3では，BGグラフィックを作成する機能に，“BG TOOL”という名前をつけ，BASICから命令で直接呼び出せるようになっています。また，BGグラフィックの画面をカセットテープに保存（セーブ）したり呼び出し（ロード）たりする場合も，BASICの命令ひとつで直接実行できるようになりました。さらにプログラムを電池でバックアップするときに，

ＢＧグラフィックも一緒にバックアップできるようになりましたから，作りかけのゲームを保存しておきたいときなどとても便利です。

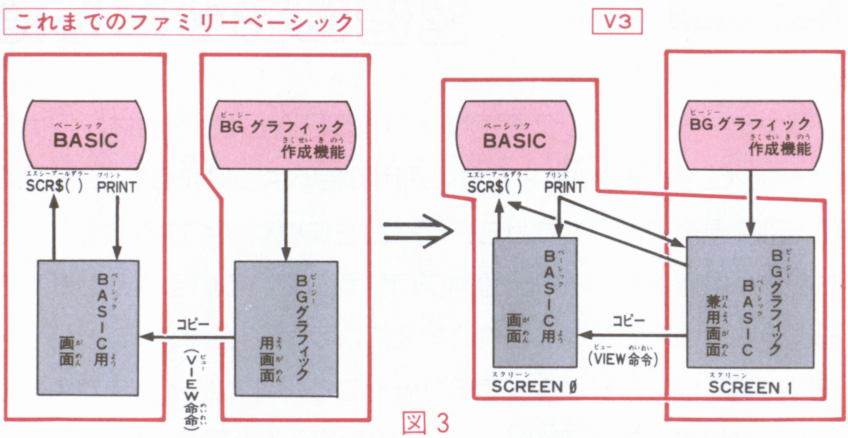


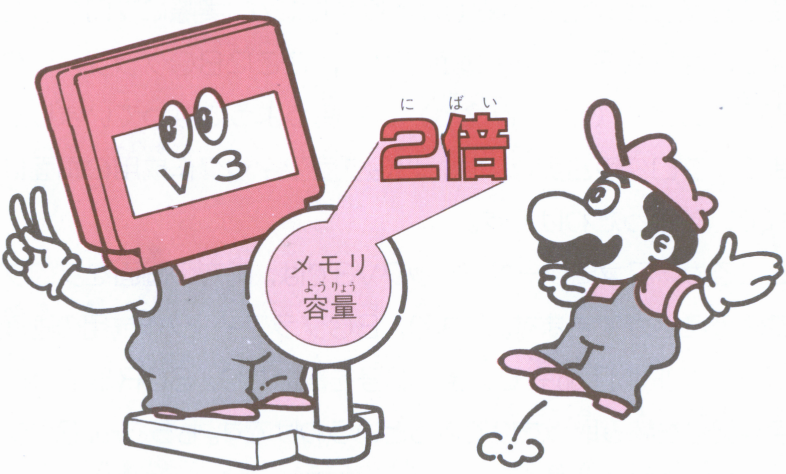
図３

ＢＧグラフィック関係でもうひとつ強化された点は，ＢＧグラフィックが２面使えるようになったことです。今までのファミリーベーシックでも，「ＢＧグラフィックを作るときの画面」と「ＢＡＳＩＣで使う画面」という2つの画面がありましたが，実際にプログラムで使えるのはＢＡＳＩＣ用の画面だけで，ここにＢＧグラフィック用の画面をコピー（ＶＩＥＷ命令）して使うようになっていました。つまり，プログラム実行中は，ＢＧグラフィック作成用の画面は使われていなかったわけです。

ところが，ファミリーベーシックV3では，両方の画面をプログラムで使うことができます。つまり，ＢＧグラフィック作成用の画面に対しても，ＰＲＩＮＴ命令で文字や記号を書いたり，ＳＣＲ＄で画面の文字や記号を読み取ったりできるというわけです。もちろん，ＢＧグラフィック作成用の画面を，もう一方の画面にコピーする機能も残っていますから，今までの使い方と併せていろいろと応用できるでしょう。

1-3 メモリ容量が2倍になりました

ファミリーベーシックで使われているBASIC言語は，ファミリーコンピュータの特徴を充分に活用できるように作られた，特別なBASIC言語です。ですから，ゲームを作りやすくする命令などがたくさん用意されており，短いプログラムでいろいろなことができるようになっています。これに加えて，ファミリーベーシックV3では，メモリの容量が今までの2倍となり，4Kバイトまで使えるようになりました。一般のパソコンと比べると少ないメモリですが，独特な特徴ある命令を活用することによって，短くて効率のよいプログラムを作ることができます。



1-4 BASICが強力になりました

ファミリーベーシック V3のBASIC言語は，これまでのファミリーベーシックをさらに拡張したもので，新しい機能がたくさん追加されています。新しく追加された命令は，全部で26種類あります(図4参照)。各命令の文法や具体的な使い方については，「4．命令早わかり」(P.95）で解説してありますので，ここでは，どんな機能が追加されたのかを，簡単に紹介しておきましょう。

●プログラム作りを助ける命令（11種）

RENUM・AUTO・DELETE
TRON・TROFF・FIND
ON ERROR GOTO・RESUME
E R L・E R R ・ ERROR

●ゲーム作りに役立つ命令（6種）

CAN ・CRASH(　)・VCT(　)
SCREEN・CLS
FILTER

●B G グラフィックの命令（5種）

BGTOOL・
BGGET・BGPUT
SAVES・LOADS

●その他の命令（4種）

BACKUP・CLICK ON/OFF
INSTR・GAME

図4

A．プログラム作りを助ける命令

まずは，プログラムを新しく打ち込んだり，打ち込んだプログラムのテストをしたりするときに便利な命令を紹介しましょう。たとえば，プログラムを打ち込むときに，行番号を自動的につけてくれるAUTO命令や行を追加・削除して，行番号がふぞろいになった場合にそろえ直すRENUM命令など，使い慣れるとなかなか便利なものです。また，プログラムの動きをテストするTRON/TROFF命令や，リストの中から必要な命令や変数名を探し出すFIND命令なども追加され，さらに使いやすくなりました。

●RENUM

すでに打ち込んであるリストの行番号を，10，20………というように，きれいにそろえ直す命令。



●AUTO

プログラムを打ち込むときに，行番号を自動的に付けてくれる命令。

●DELETE

すでに打ち込んであるリストの，何行分かをまとめて消す命令。

●TRON／TROFF

プログラムを実行させたとき，リストの行番号を実行順に表示させる命令と，実行を元の状態に戻す命令。

●FIND

指定した文字列をリスト中から探し，該当する行を表示させる命令。「○○という命令を使っている行をリストアップする」といっ

た使い方ができます。

●ON ERROR GOTO／RESUME

プログラムを実行させたとき，「エラーが起きたら○○行にジャンプする」という指定をしておく命令。そして，ジャンプした後でもう1度元の行に戻すために，RESUME命令を使います。

●ERL／ERR／ERROR

プログラムの中でエラー処理をするための命令。どの行でどんなエラーが起きたかを調べたりするときに使います。

B．ゲーム作りに役立つ命令

次に，ゲームのプログラムを作るときに活用すると，効率のよくなる命令を紹介しましょう。とくに CAN，CRASH（ ），VCT（ ）などは動きが複雑で，しかも速いプログラムを作りたいという場合に，大いに力を発揮してくれます。こうした命令をたくさん持っていることが，ファミリーベーシックV３が短くて効率のよいプログラムを作れる秘密なのです。

●CAN

DEF　MOVEによる指定を取り消す命令。

●CRASH（ ）

DEF MOVEで動かしたマリオなどのアニメキャラクタが，他のアニメキャラクタとぶつかったかどうかを調べる命令。レーザーの命中判定などに使います。



●VCT（　）

MOVE命令で移動中のアニメキャラクタの移動方向を調べる命令。

●SCREEN

BG面∅とBG面1を切り替えて使うための命令。片方を表示しながら，もう一方の画面に書き込むといったこともできます。

●CLS

今までのCLS命令を拡張し，2つの画面のうちどちらか指定した方だけを消すことができる命令。

●FILTER

画面全体を淡く着色する命令。

C．BGグラフィックの命令

BGグラフィックは，ファミリーベーシックの大きな特徴のひとつで，プログラムとは別に背景画を描けることにより，メモリを大幅に節約できるようになっています。ファミリーベーシックV3では，さらにこの機能を使いやすくするために，BGグラフィックの画面を電池でバックアップしたり，BASICからの命令で，カセットテープに保存したり呼び出したりすることができるようになりました。ただし，これらの命令でバックアップや保存ができるのは，BG面1（BGグラフ



イック作成用の画面）だけですから注意してください。

●BGTOOL

BASICから，BGグラフィックの作成機能を呼び出す命令。

●BGGET／BGPUT

BG画面を，電池でバックアップするための準備をする命令と，バックアップしたBG画面を元に戻す命令。これらの命令を忘れるとBG画面のバックアップができません。

●SAVES／LOADS

BG画面をカセットテープに保存したり，呼び出したりする命令。プログラムの保存・呼び出しは従来のSAVE／LOADですから，この命令はこれにSCREENのSをつけてあるわけです。なお，プログラムとBG画面を同時にSAVE／LOADする方法もあります（P.24参照）。

D．その他の命令

●BACKUP

プログラムやBG画面を電池でバックアップするために，実行する命令。この命令を忘れると，正しくバックアップができません。

●CLICK ON／OFF

キーを押したときに，「ピッピッ」という音（クリック音）を出す，出さない，という指定をする命令。

●INSTR（ ）

データ文字列の中に，指定した文字列があるかどうかを調べる命令。

●GAME

内蔵されている4つのプログラムを呼び出すための命令。呼び出すと，すぐに実行できます。プログラムの番号は0～3で，一緒に内蔵されている背景画面の呼び出しも指定できます。

※キーボードの使い方，命令，キャラクタテーブルA・B，カラーチャート等については，これまでのファミリーベーシック「取扱説明書」をご覧ください。



2. FAMILY BASIC™ V3の基本操作



2-1 接続のしかた

基本的な接続方法は，これまでのファミリーベーシックと変わりありません。

ただし，ファミリーコンピュータ本体に差し込むカセットが，これまでのものからV３カセット(赤)に替わります。

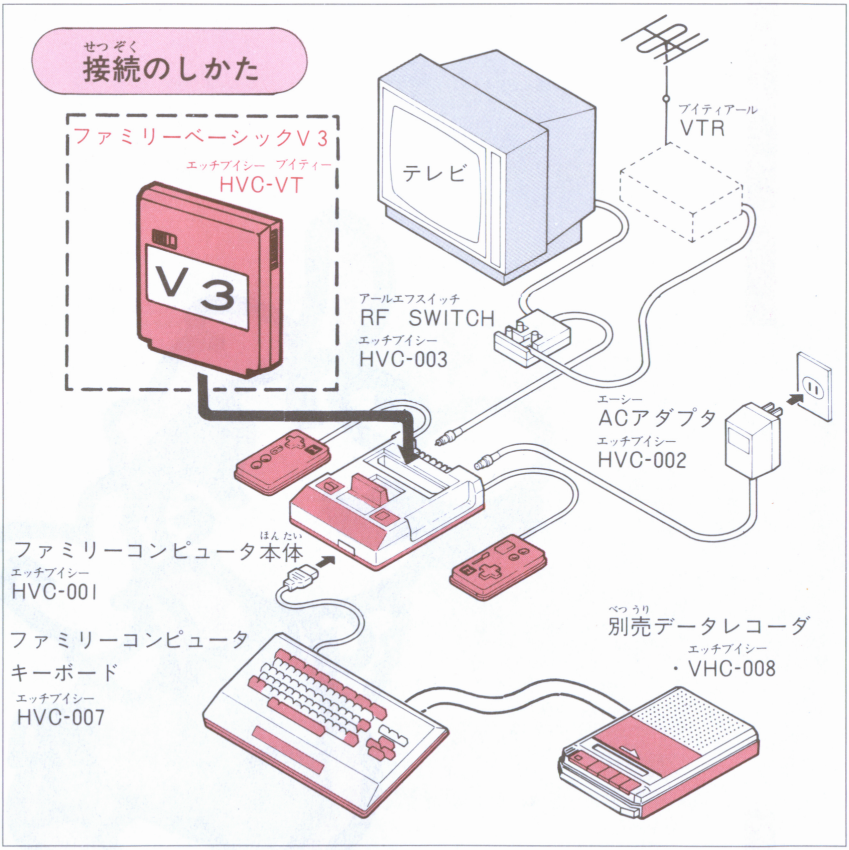


図５

2-2 電源の入れかた

①キーボードのコネクタを差し込む

電源(POWER)をOFFにした，ファミリーコンピュータのエキスパンドコネクタ（機能拡張コネクタ）にキーボードのコネクタを接続します。

②V３カセットを挿入する

V３カセットを，ファミリーコンピュータのカセットコネクタに差し込みます。

③電源をONにする

ファミリーコンピュータの電源(POWER)を入れます。

④BASIC準備よし！

BASＩCが呼び出され，ファミリーコンピュータの準備が整うと，OKと画面に表示されます。

このとき，OKと表示されずに，

１.)「キーボード　ヲ　セツゾク　シテクダサイ」

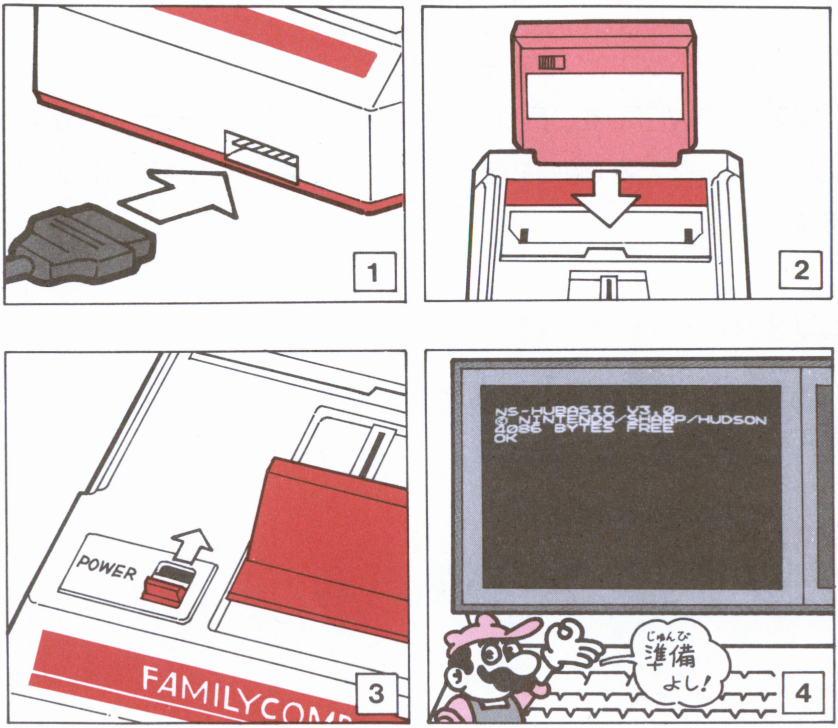
とメッセージが出たら，キーボードのコネクタがはずれていないかどうか，もう１度よく確かめてください。また，

２.)「バックアップ　スイッチ　ヲ　OFF　ニ　シテクダサイ」

とメッセージが出たら，V３カセットの左上部分にあるバックアッ

プスイッチが，ONになっていないかどうかを確かめてください。

以上の手順を，図にしておきますので参照してください。



このようなスタート方法を，COLD STARTと言います。ふつうは，このCOLD STARTで操作を開始します。ただし、この方法はすべて新しい状態ですから，ファミリーコンピュータは何もデータを持っていません。この COLD START に対して，HOT START という方法があります。この方法は，メモリバックアップのときに使用しますので，詳しくは「2-4. メモリバックアップについて」(P.19) をお読みください。

2-3 内蔵プログラムの使い方

V3カセットには，4つのプログラムと1つの背景画（BG面）が内蔵されています。これらは，画面にOKと表示されているときならば，いつでもプログラムを呼び出すことができます。

プログラムを呼び出す方法は，ファンクションキーを押す方法とコマンドを入力する方法の2とおりがあります。

Ⓐファンクションキーで呼び出す方法

4つのプログラムが，ファンクションキーF1～F4にそれぞれ対応しています。呼び出したいプログラムのキーを押せば，すぐに始めることができます。

Ⓑコマンドを入力して呼び出す方法

コマンドを入力する場合は，GAME命令を使います。

GAME　P,Sw

Pはゲーム番号で，Ø～3のいずれかを選択します。またSwは背景画をカセットから転送するかどうかの指定で，転送する場合は1，しないならØまたは省略します。たとえば，GAMEØを選択する場合は，

GAME Ø RETURN

とキーを押せばよいのです。



①内蔵プログラムを呼び出すことによって，前に記憶されていたプログラムは自動的に消えてしまいますので，注意しなくてはなりません。残したいときは，カセットテープに保存しておきましょう。

②背景画だけを転送することもできます。その場合は，

GAME , 1 RETURN

とキーを押します。

③4つの内蔵されているプログラムと1つの背景画は，どんな操作をしても消えることはありません。

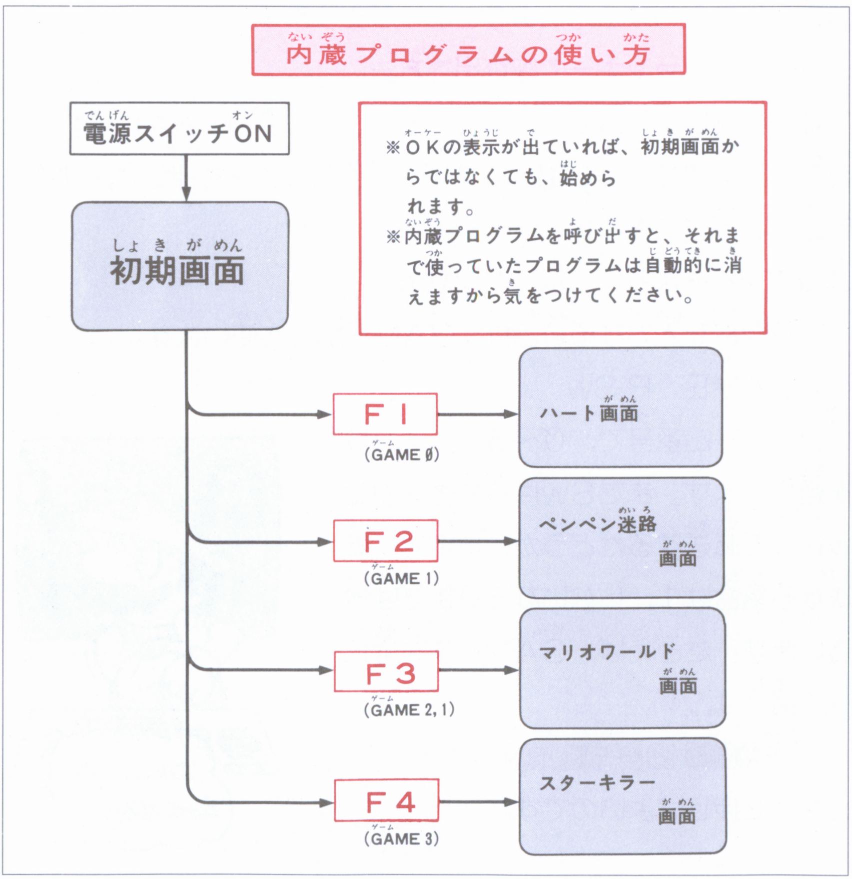


図6

2-4 メモリバックアップについて

BASICのプログラムやBGグラフィックの画面は，単三乾電池を利用して，一時的にV３カセット内に記憶(メモリバックアップ)しておくことができます。プログラムを作っている途中で，電源を切ってひと休みする場合などに利用してください。

A．プログラムとBGグラフィックの両方をメモリバックアップする場合

この場合には，最初にBGGET命令でBGグラフィックを保存する準備をし，その後でBACKUP命令を使い，画面の指示に従ってバックアップスイッチをONにします。BGGET命令の後でBACKUP命令という順番を，まちがえないように気をつけてください。



①BGGET RETURN BACKUP RETURN とキーを押します。

②「バックアップ　スイッチ　ヲ　ON　ニ　シテクダサイ」とメッセージが表示されます。

③V３カセット左上のバックアップスイッチをON（右側）にします。

④「デンゲン　ヲ　OFF　ニ　シテクダサイ」
とメッセージが表示されます。

⑤本体の電源をOFFにし，V3カセットを抜いて終わりです。

なお，BGGET命令を入力すると，"OM ERROR" と出ることがあります。これは，「プログラムとBGグラフィックの両方をバックアップするにはメモリが足りない」という意味です。この場合は，どちらか一方をカセットテープに保存するなどして，残りの片方だけをバックアップしてください。

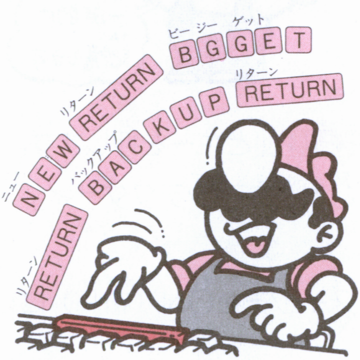
B．プログラムだけを メモリバックアップする場合

基本的には両方バックアップする場合と同じですが，BGグラフィックのバックアップ準備が不要となり，手順は次のようになります。

①B A C K U P RETURN とキーを押します。

②以下の手順は，Aとまったく同じです。

C．BGグラフィックだけを メモリバックアップする場合



これも，基本的には両方バックアップする場合と同じです。しかしこの場合は，プログラムが不要ということなので，両方バックアップする場合の手順①を次のように変更し，バックアップする前にプログラムを消してしまいます。

この場合，１度消したプログラムは元に戻せませんので，残しておきたいプログラムはカセットテープに保存しておくようにするとよいでしょう。

①NEW RETURN BGGET RETURN BACKUP RETURN
とキーを押します。

②以下の手順は，Aとまったく同じです。

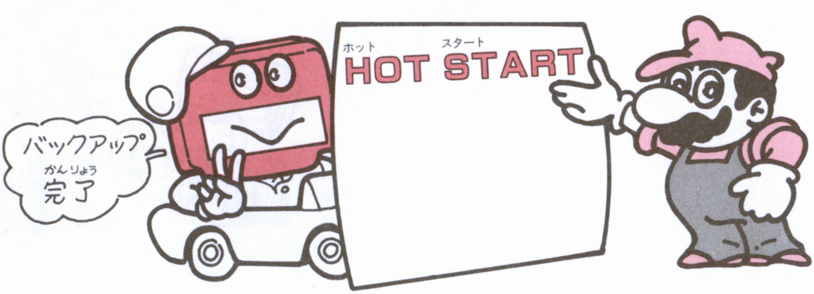
D．メモリバックアップしたものを使える状態に戻す手順

①V３カセットを本体に差し込み，本体の電源スイッチをONにします。

②「バックアップ　スイッチ　ヲ　OFF　ニ　シテクダサイ」とメッセージが表示されます。

③V３カセット左上のバックアップスイッチをOFF（左側）にします。

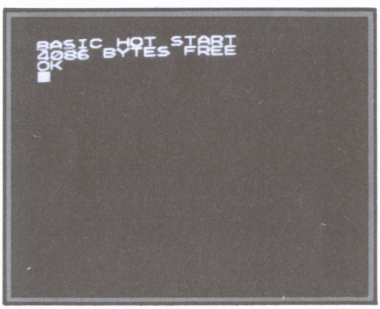
④BASICの状態になり，記憶されていたプログラムやBGグラフィックが使えるようになります。



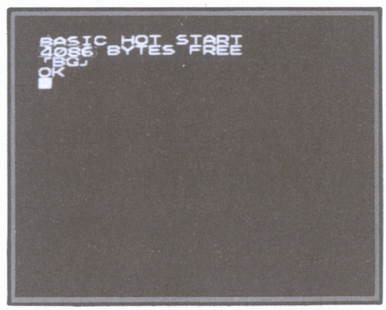
なお，バックアップスイッチをONにした状態で，本体の電源スイッチをONにしてスタートすることをHOT STARTといい，画面にも「HOT START」と表示されます。これは，すでにプログラムがバックアップされている状態です。

また，BGグラフィックがある場合は，「HOT START BG」と表示されます。そして，自動的にユーザーRAMからVRAMへ，BG面1のデータが転送され，BGPUT命令の実行と同じ状態になります。

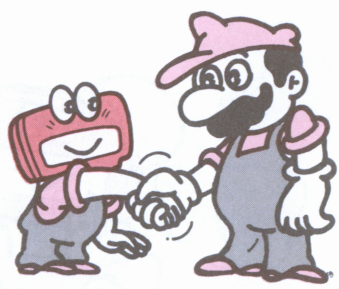
このとき，もう1度BGPUT命令を実行しても，データはすでに転送済みのため，NBエラーが発生します。



HOT START



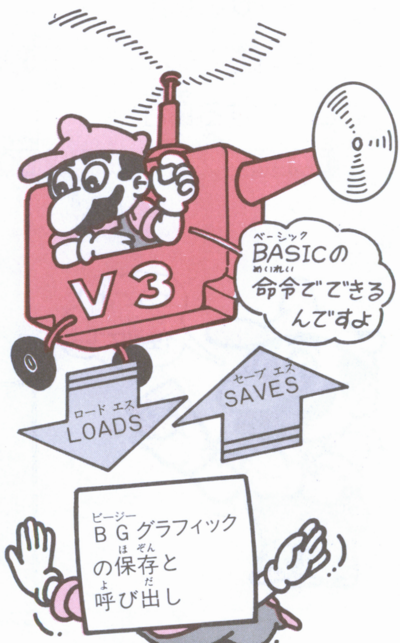
BGグラフィックがある場合



2-5 カセットテープへの保存と呼び出し

ファミリーベーシックＶ３では，これまでのＳＡＶＥ／ＬＯＡＤ（プログラムの保存／呼び出し）命令の他に，ＳＡＶＥＳ／ＬＯＡＤＳ（ＢＧグラフィックの保存／呼び出し）という２つの命令が追加されました。この命令によって，これまでＢＧグラフィック作成機能を使わないとできなかった画面の保存が，ＢＡＳＩＣの命令でできるようになりました。それぞれの命令の使い方は，プログラムのＳＡＶＥ／ＬＯＡＤと同じです。（ファミリーベーシック取扱説明書の「プログラムのセーブ（保存），ＬＯＡＤ（呼び出し）」P.45～48）

また，これまでのファミリーベーシックで作って，カセットテープに保存してあったＢＧグラフィックも，この命令を使って読み込むことができます。



●内蔵プログラムについて●

Ｖ３カセットに内蔵されている４つのプログラムも，自分で打ち込んだプログラムと同じように，ＳＡＶＥ命令でカセットテープに保存することが

できます。もちろん，いろいろな改造をした後で，SAVEすることもできます。

また，SAVES／LOADSで画面の保存／呼び出しもできますから，『マリオワールド』の画面をいくつも作っておいて，使い分けるというような応用もできるわけです。

● プログラムとBGグラフィックを一緒にSAVE/LOADする方法

プログラムとBGグラフィックを組み合わせて，1つのゲームになっているものなどは，SAVE／LOADするときも，一緒になっていたほうが便利です。次のようにすれば，プログラムとBGグラフィックを，同時にSAVE／LOADすることができます。

①セーブする場合

SAVE"プログラム名":SAVES"BGのファイル名" RETURN

②ロードする場合

LOAD"プログラム名":LOADS"BGのファイル名" RETURN



※V3カセットでの，スタートから終了までの全体の流れ，操作手順はP.121～125で図解しています。

3. FAMILY BASIC™ V3のゲームプログラム

ファミリーベーシック



3-1 プログラムの改造方法

ファミリーベーシックV3には，ワンタッチで呼び出せる4つのプログラムが内蔵されています。呼び出したプログラムは，そのままでも遊べますし，BASICで作られていますから，自分で改造することもできます。

改造で必要となるリストの変更方法を説明しておきましょう。

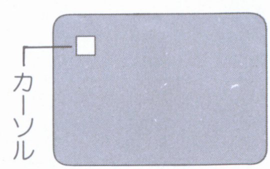
■必ず，次の操作をした後に，改造を始めてください。

1．STOP キーを押してプログラムを止めます。

2．SHIFT ＋ CLR HOME キーを押すと，リストが消え，カーソルがホーム位置に戻ります。

3．CTR ＋ D キーを押してアニメキャラクタを消し，文字の色を白色に戻します。

4．リスト命令で，必要なプログラム行を呼び出します。



画面

① 1文字変更したいとき

100X=2　を　100X=3

のように，1文字だけ変更したいときは，

```
100 X=2
      ↑
    カーソル
```

カーソルを「2」の上に動かして，3キーを押してから RETURN キーを押してください。

②　間に何文字かを入れたいとき

1ØØ　A＝SCR$(X，Y)

を

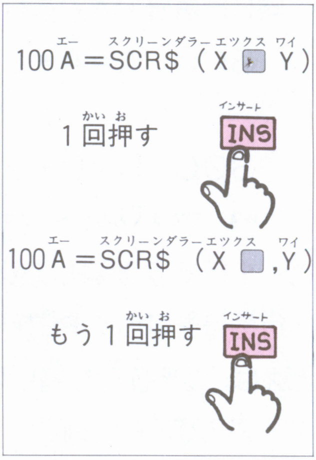
1ØØ　A＝SCR$(X＋1，Y)

のように変更したいときは，

1ØØ　A＝SCR$（X■Y）

カーソル

カーソルを「，」の上に動かし，INS INS ＋ 1 RETURNとキーを押します。INSキーを押すと，押した回数だけ間があきます。



③　何文字かを詰めたいとき



1ØØ　PRINT　SC＋ST

を

1ØØ　PRINT　S＋ST

のように変更したいときは，

1ØØ　PRINT　SC■ST

カーソル

カーソルを「＋」の上に動かしてDELキーを押してください。その後RETURNキーを押します。押した回数分だけ，カーソルが左側の文字を消していきます。

④　新しい行を追加したいとき

画面に「OK」が表示されて，その下にカーソルが出ているときに，追加したい行だけを行番号を付けて入力してください。入力を終えてからリストを出せば，追加した行も含んで行番号順に出てきます。

⑤ 1行を全部変更したいとき

同じ行番号で入力すれば，あとから入力したものに変わります。

```
300 A=A+10 : GOTO 500
```
←前の行を消さなくても

↓

```
300 B=B+20 : GOSUB 650
```
←同じ行番号で入力すれば，これに変わります。

※④，⑤では入力した後，必ず RETURN キーを押してください。

⑥ 1行を全部消したいとき

行番号を打って RETURN キーを押せば，その行が消えます。たとえば，250行を消したいならば，

2 5 0 RETURN

とキーを押してください。

⑦ 行と行がつながってはいけません

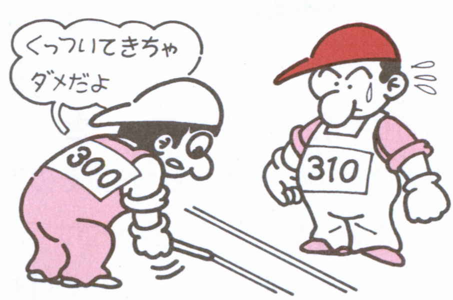
プログラム中で，ある行と次の行がつながってしまうと，エラーになったり，正しく動かないことがあります。

ある行が画面のいちばん右端で終わっているときは，リストを出すと次の行との間が1行あきます。たとえば，

```
300 IF X<8 AND Y<8 GOTO 1000

310 X=X+AX : Y=Y+AY
```
1行あく

のように，300行が画面の右端で終わりなら，リストを出したとき，300行の下に1行あいて310行が出てくるのが正しいのです。これが，



```
300 IF X<8 AND Y<8 GOTO 1000
310 X=X+AX : Y=Y+AY
```

と出てきたときは，300行と310行は，つながっていることになります。このようなときは，もう一度正しく入力し直してください。

⑧ 内蔵プログラムを改造する前に

F1 F2 F3 F4 の4つのキーは，ワンタッチで内蔵ゲームを呼ぶことができる便利なキーですが，プログラムの改造をしている最中にこのキーを押すと，改造したプログラムがアッという間に消えてしまいます。これを防止するには改造したいプログラムを呼び出した後

K E Y　1, " " RETURN

K E Y　2, " " RETURN

K E Y　3, " " RETURN

K E Y　4, " " RETURN

とキーを押してください。次に，

K E Y F7

（または K E Y L I S T RETURN）

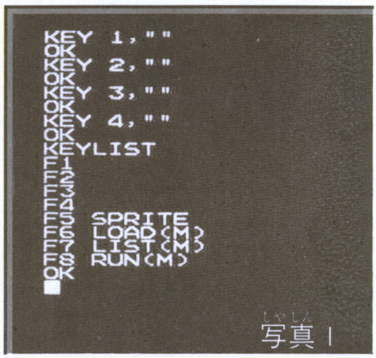


写真1

とキーを押して，写真1のように，F1～F4の右側に何も出ないことを確かめてください。こうしておけば，まちがって F1～F4 のキーを押してしまっても，改造中のプログラムが消えてしまうことはありません。

F1～F4のキーを元に戻すには，ファミリーコンピュータ本体のリセットキーを押します。ただしリセットキーを押すと本体のプログラムはそのまま残りますが，BGグラフィックは消えてしまいます。BGグラフィックを残したいときは，BGグラフィックをテープにセーブしてから，リセットキーを押してください。

3-2 内蔵プログラムで遊ぼう

1. ハート

A. 遊び方

プログラムを呼び出すときは，F1キーを押すか，または，

GAME 0 RETURN

とキーを押してください。プログラムが始まると，音楽が流れ，

PLAY ? Y/N=>■ （YはYes, NはNoの意味です）

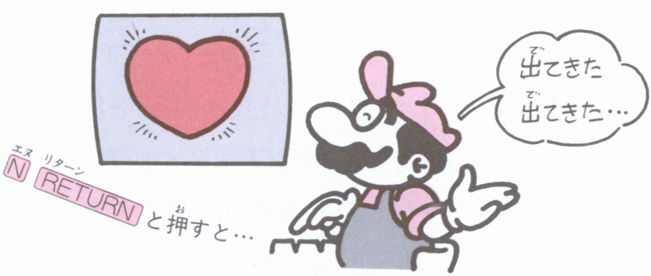
とメッセージが出てきます。 ↑カーソル

N RETURN

とキーを押してください。こうするとハートが出てきます。遊びたくなったらSTOPキーを押してプログラムを止めてから，F8キーを押すか，

RUN RETURN

とキーを押せば，メッセージが再び出てきます。



ここで

Y RETURN

とキーを押すと画面は暗くなり，ハートを描く準備ができました。そこで，IIコントローラのVOLUME（ボリューム）のつまみをいちばん右まで動かしてから，マイクの左上の小さな穴に向かって息を吹きかけるか，声を出してください（図7）。

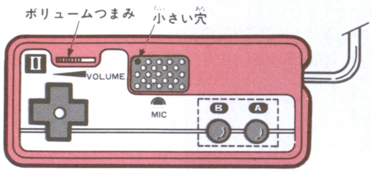


図7

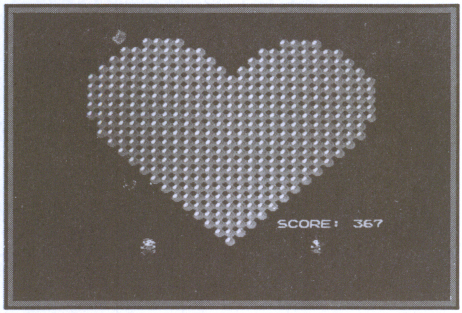


写真2

息を吹きかけると，ハートのマークが少しずつできてきます。次にマリオとレディーが画面の下に出てきて，真ん中で出合うと音楽が流れます（写真2）。終わるとまた初めのメッセージ画面に戻りますので，くり返して遊べます。終わりたいときは，STOP キーを押してください。

B．改造の方法

① ハートの色を変えてみましょう

ハートを描いているときの色は，80行で決めています。

80 SPRITE ON：CGSETØ，Ø：U=&H15

「&H15」は16進数で，10進数の数字にすると「21」のことです。この数字をほかの数字に変更すると，ハートの色が変わります。

※文中に出てくる「&」は，「&」と同じ文字です。画面に出てくる文字と多少異なりますので，ご了承ください。

何色になるのかは，図8の色コード表にでています。たとえば「U＝39」にすると黄色のハートになります。

《52色の色コード》

	16進	10進	16進	10進	16進	10進	16進	10進	
	ØØ	Ø	1Ø	16	2Ø	32	3Ø	48	灰色～白色
青系	Ø1	1	11	17	21	33	31	49	有色
青系	Ø2	2	12	18	22	34	32	5Ø	有色
青系	Ø3	3	13	19	23	35	33	51	有色
青系	Ø4	4	14	2Ø	24	36	34	52	有色
赤系	Ø5	5	15	21	25	37	35	53	有色
赤系	Ø6	6	16	22	26	38	36	54	有色
赤系	Ø7	7	17	23	27	39	37	55	有色
赤系	Ø8	8	18	24	28	4Ø	38	56	有色
緑系	Ø9	9	19	25	29	41	39	57	有色
緑系	ØA	1Ø	1A	26	2A	42	3A	58	有色
緑系	ØB	11	1B	27	2B	43	3B	59	有色
緑系	ØC	12	1C	28	2C	44	3C	6Ø	有色
	ØD	13	1D	29	2D	45			クロ
	ØE	14	1E	3Ø	2E	46			クロ
	ØF	15	1F	31	2F	47			クロ

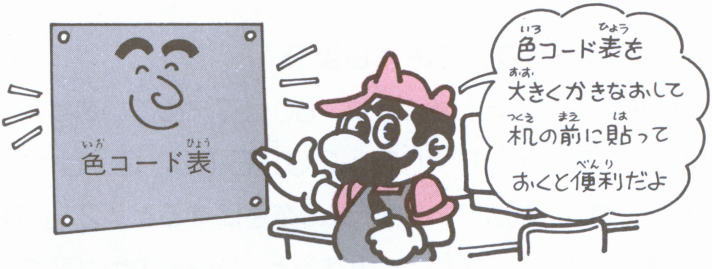
色相（縦軸）　暗←→明（横軸）

図8

音楽が流れているときのハート色は，43Ø行の，

430　W=&H34：FOR　N=ØTO1：……以下略

で決まります。「&H34」は，10進数の数字」では「52」のことです。色は淡いピンクです。この数字も色コード表を見ながら，変えてみましょう。



※文中に出てくる「&」は，「&」と同じ文字です。画面に出てくる文字と多少異なりますので，ご了承ください。

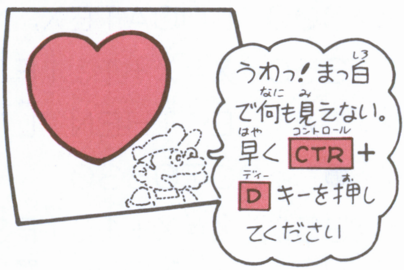
② バックの色を変えてみましょう

バックの色は黒ですが，この色は90行のPALET B 命令で決めています。

90　PALETB　Ø，15，48，48，U

この数字も色コード表を見ながら変えてみましょう。たとえば「1Ø」にすると，バックの色が緑になって，草原のようになります。音楽が流れているときのバックの色は，480行で赤に決めています。

バックを白にすると…



48Ø　LOCATE1Ø，1Ø：PRINT "PERFECT"：FORN=ØTO1ØØ：PALETB　Ø，6，&H3Ø，……以下略

この「6」も，ほかの数字に変えてみましょう。

バックの色を変えるときは，ハートの色と同じにならないように注意してください。ハートが見えなくなってしまいます。また，バックの色を白にすると，STOPキーを押して中断したときに，文字が見えなくなってしまいます。このようなときは，

CTR＋D

（CTRキーを押しながらDキーを押します）

とキーを押せば，文字が見えるようになります。



※文中に出てくる「&」は，「&」と同じ文字です。画面に出てくる文字と多少異なりますので，ご了承ください。

③ ハートを描くキャラクタを変えてみましょう

ここでは，ハートを丸いボールで描いていますが，何かほかのキャラクタで描いてみましょう。それには，まずプログラムの3カ所を，次のように変えてください。

```
3ØØ LOCATEX，Y：PRINT "●"；：…以下略
3ØØ LOCATEX，Y：PRINT H$；：…以下略

35Ø X=X+1：IFSCR$（X，Y）＝" " AND
  X<=E THEN LOCATE X，Y：PRINT "
●"：GOTO33Ø
35Ø X=X+1：IFSCR$（X，Y）＝" " AND
  X<=E THEN LOCATE X，Y：PRINT
H$：GOTO33Ø

41Ø X=-1：FORC=S TOE：IFSCR$（C
，Y）＝"●" AND………以下略
41Ø X=-1：FORC=S TOE：IFSCR$（C
，Y）＝H$ AND………以下略
```

この3カ所を変えた後，次の1行を新しく追加してRUNしてみましょう。

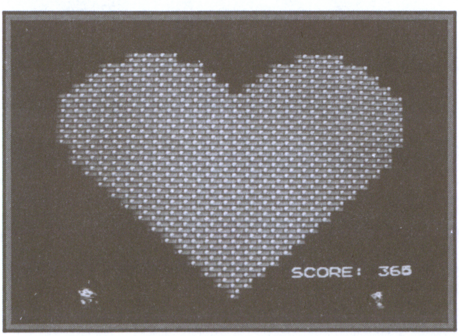


写真3

135　H$=CHR$(195)

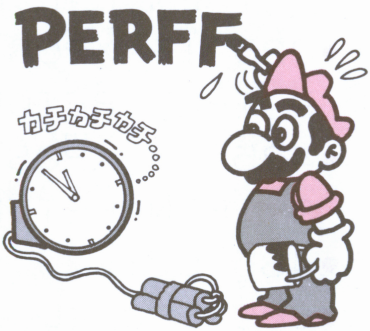
すると，ハートがレンガで描かれるようになります（写真3）。

新しく追加した135行のCHR$（　）の中の数字を変えると，いろいろなキャラク



タでハートを描くことができます。184～255の間の数字にすれば，BGグラフィックで使うキャラクタになり，33～183にすると文字や記号になります。どのキャラクタになるのかは，本書付録（P.134～135）のキャラクタコード表Bを見てください。なお，キャラクタの種類によっては，色が変わらないことがあります。

④　タイマーを付けてみましょう



ハートのプログラムは，PERFECTの文字が現れるまで実行されますが，タイマーを作って制限時間を設けましょう。

タイマーを付けるためには，いくつかの行を変えたり，追加しなければなりません。

〈次のように行を変えてください〉

```
230 IF(PEEK(&H4Ø16) AND &H4)
=Q THEN23Ø
      ↓
23Ø IF(PEEK(&H4Ø16) AND &H4)
=Q THEN225

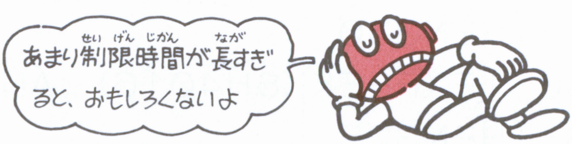
33Ø IF(PEEK(&H4Ø16) AND &H4)
=Q THEN33Ø
      ↓
33Ø IF(PEEK(&H4Ø16) AND &H4)
=Q THEN325
```

※文中に出てくる「&」は，「&」と同じ文字です。画面に出てくる文字と多少異なりますので，ご了承ください。

```
44Ø IF (PEEK (&H4Ø16) AND &H4)
=Q THEN44Ø
44Ø=IF (PEEK (&H4Ø16) AND &H4)
=Q THEN435
```

〈次に新しい行を追加してください〉

```
145 DEF MOVE (7) =SPRITE (11, 3,
25, 255, Ø, Ø) :POSITION 7, 16, 2Ø
:MOVE 7
●225 IF XPOS (7) >216 AND Q<>99
9 THEN P=L:NEXT:RETURN 6ØØ
●325 IF XPOS (7) >216 AND Q<>99
9 THEN RETURN 6ØØ
●435 IF XPOS (7) >216 AND Q<>99
9 THEN RETURN 6ØØ
●485 CUT 7
●6ØØ ' ジカンギレ
●61Ø CGSET1, 1 :CUT 7:PLAY "O2T1
B3AGFEDC"
●62Ø LOCATE 1Ø, 7 : PRINT " ジカ
ンギレ "
●63Ø PAUSE 3ØØ: ERA 7
●64Ø GOTO 19Ø
```



※文中に出てくる「&」は，「&」と同じ文字です。画面に出てくる文字と多少異なりますので，ご了承ください。

このように改造すると，画面の上の方にニタニタのタイマーが出てきます。はじめニタニタは左の方にいて，少しずつ右に動いて行き，いちばん右まで行くと，時間ぎれになって終わります。



ニタニタの動く速さを変えると，制限時間を変えることができます。新しく追加した行のDEF MOVE命令の中に

```
145 DEF MOVE (7)=SPRITE(11, 3,
25, 255, Ø, Ø) :………以下略
```

25という数字があります。この数字を大きくすると制限時間が長くなり，小さくすると短くなります。でもあまり大きくしたり小さくしたりすると，タイマーの意味がなくなってしまいますから注意してください。

C．マイクを使ってみましょう

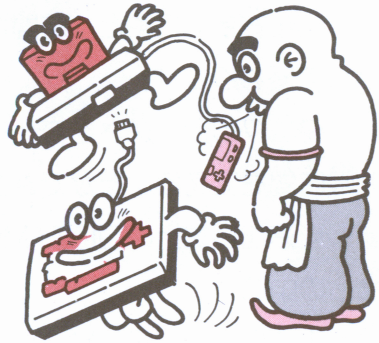
① マイクの使い方



IIコントローラにはマイクが付いています。このマイクに音や声が入ったかどうかは，BASICで知ることができます。これには，PEEK( )命令とメモリ&H4Ø16番地を使用します。では，次のような1行のプログラムを入力して，RUNしてください。

※文中に出てくる「&」は，「&」と同じ文字です。画面に出てくる文字と多少異なりますので，ご了承ください。

```
1 PRINT PEEK (&H4016) AND&H4 : G
  OTO 1
```



RUNすると，画面に数字が次々と出てきます。そこで，VOLUMEをいっぱいにしてからマイクに向かって息を吹いたり声を出すと，その数字が変わります。数字が変わったときが，マイクに音が入ったときなのです。

これをプログラムで使うには，はじめに PEEK（&H4016）AND&H4 を変数に代入しておき，マイクからの音入力で PEEK（&H4016）AND&H4 の値と同じかどうか，判定します。このテクニックを使ったプログラムの例を次にあげておきましょう。

② 肺活量遊び

このプログラムは，一息がどのくらいの長さになるかを数字で表しています。RUN するとメッセージが出て，スタートの音がします（写真4）。

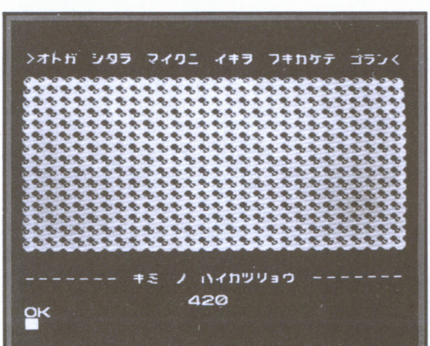


写真4

すぐにマイクに向かって息を吹いてください。息がとぎれると，そこで数字が出てきます。大きな数字にしたいのなら，そうっと長く吹きかけるのがコツです。もう一度始めたいときは，RUN してく

※文中に出てくる「&」は，「&」と同じ文字です。画面に出てくる文字と多少異なりますので，ご了承ください。

ださい。ここに出てくる数字は学校などで測る肺活量と同じではありません。また，画面に出てくるBGキャラクタの種類と色は，RUNのたびに変わります。

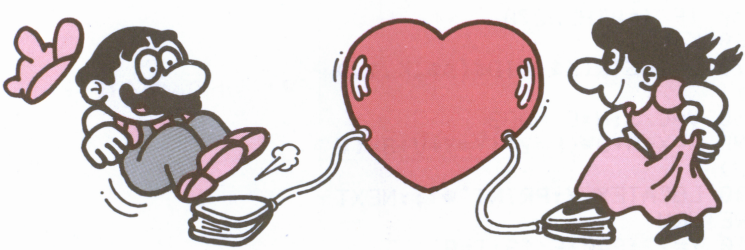
```
100 'ﾊｲｶﾂﾘｮｳ ｹﾞｰﾑ
110 CLEAR:CLS:C=0:F=0:CP=184
+RND(72)
120 PV=PEEK(&H4016)AND&H4
130 PALETB 0,14,48,48,17+RND
(12)
140 PRINT ">ｵﾄｶﾞ ｼﾀﾗ ﾏｲｸﾆ ｲｷ
ｦ ﾌｷｶｹﾃ ｺﾞﾗﾝｸ"
150 PAUSE 200 : PLAY "V15O3T
1C1DEFGABO4C"
160 IF (PEEK(&H4016)AND&H4)=
PV GOTO 160
170 FOR SS=0 TO 19
180 S=PEEK(&H4016)AND&H4
190 IF S<>PV THEN SS=19:F=1


200 NEXT
210 IF F<>1 GOTO 240
220 C=C+1:PRINT CHR$(CP);
230 IF S<>PV THEN F=0:GOTO 1
70
240 PRINT:PRINT
250 PRINT "-------- ｷﾐ ﾉ ﾊｲｶﾂ
ﾘｮｳ --------"
260 PRINT "          [redacted]";C;"
[redacted]"
270 END
```

← 変数PUに入れておく（130行）

← 息が吹きかけられるまでこの1行だけを実行する（160行）

← 息が続いている間は170行に戻る（230行）



ハート　プログラムリスト

```
10 PLAY"M1V9Y2T3:M1V7Y1T3:M1
T3"
20 PLAY"O3E5G:O3C5C:O2G5G"
30 PLAY"O4CO3B4A1:CC:#F#F"
40 PLAY"G4A1:F4F1:O3D4O2B1"
50 PLAY"B4O4C7:F4E7:G4C7"
60 CLS:LOCATE 8,10:Q=PEEK(&H
4016) AND &H4:INPUT"PLAY ? Y
/N=>",Y$:IF Y$="N" THEN Q=99
9
```

●音楽を流して遊ぶかどうかを選ぶ

```
70 CLS:ERA0,1,2,3,4,5,6,7
80 SPRITE ON:CGSET0,0:U=&H15

90 PALETB 0,15,48,48,U
100 A$="AIIIIIIIIIIIGIGGGGH
"
110 B$="HHBHBBBEBEEE"
120 K$="ABCDEFGHI"
130 I$="ABCGHIDEF"
```

●画面の色を決めるなどの準備

```
140 CLS:LOCATE9,10:PRINT"STA
RT !!":PAUSE80:CLS
150 X=13:Y=22:I=0
160 GOSUB220:SWAPA$,B$:GOSUB
220
170 IF I=0THENSWAPK$,I$:I=1:
GOTO160
180 GOSUB310:GOSUB430
190 IF Q<>999THEN60
200 GOTO70
210 END
```

●メインルーチン
（サブルーチンを呼ぶ）

```
220 L=LEN(A$):FORP=1TOL
230 IF(PEEK(&H4016) AND &H4)
=Q THEN230
240 K=P
250 IF I=0THEN270
260 K=L+1-P
270 C=INSTR(K$,MID$(A$,K,1))
-1
280 V=C/3:H=C MOD3
290 X=X+H+3*(H=2):Y=Y+V+3*(V
=2)
300 LOCATEX,Y:PRINT"●";:NEXT
:RETURN
```

●ハートのふちを描く

```
310 Y=22:S=0:E=13:T=0
```

※本書リストに出てくる「＊」は，「＊」と同じ文字です。画面に出てくる文字と，多少異なりますので，ご了承ください。

```
320 LOCATE18,20:PRINT"SCORE:
";
330 IF(PEEK(&H4016) AND &H4)
=Q THEN330
340 T=T+1:LOCATE24,20:PRINTT

350 X=X+1:IFSCR$(X,Y)=" "AND
 X<=E THEN LOCATE X,Y:PRINT"
●":GOTO330
360 Y=Y-1:IFY<1THEN390
370 GOSUB410 :IFX=-1THEN360
380 GOTO350
390 IFS=0THENS=13:E=26:Y=20:
GOSUB410:GOTO330
400 RETURN
```

●ハートを塗りつぶす

```
410 X=-1:FORC=S TOE:IFSCR$(C
,Y)="●"AND SCR$(C+1,Y)=" "TH
EN X=C:C=E
420 NEXT:RETURN
430 W=&H34:FORN=0TO1:DEFMOVE
(N)=SPRITE(N,3+4*N,1,1,0,0):
POSITIONN,236*N,200:NEXT
440 IF(PEEK(&H4016) AND &H4)
=Q THEN440
450 T=T+1:LOCATE24,20:PRINTT

460 MOVE0,1:IFCRASH(0)=1THEN
480
470 GOTO440
```

●マリオとレディを動かす

```
480 LOCATE10,10:PRINT"PERFEC
T":FORN=0TO100:PALETB 0,6,&H
30,&H30,U:SWAPU,W:NEXT
490 PLAY"M1V2Y1T2:M1V10Y1T2:
T2"
500 PLAY"O3E3G#DGDG#DG:O1#B5
B#AB:R"
510 PLAY"EGO4CO3E:Y2O2#B6E3:
O1C5C"
520 PLAY"#GAO4DC:F5#F:#C#C"
530 PLAY"O3B#ABO4A:F3O3DFA:D
6F3"
540 PLAY"GO3AO4FO3B:E5DV15:G
A#AB"
550 PLAY"#B3GT3FGE1DV12E7:C9
:#B5T3GC7"
560 PALETB 0,15,48,48,21:RET
URN
```

●マリオとレディが出合ったとき音楽を流す

ハート変数リスト

Q	ゲームをするかしないか
X，Y	BGキャラクタを描くBG座標
U	ハートの色
W	ハートが完成し音楽が鳴っているときのハートの色
I，L，P，K， C，V，H	ハートを描く作業用
A$，B$，K$， I$	ハートの形をあらわす文字データ
N	左右から出てくるアニメキャラクタのMOVE番号
T	進度カウンター

(注) 文字データは，次のようにX，Yの変化分を表す。

Yの変化分	(Ø)			(+1)			(−1)		
Xの変化分	Ø，	+1，	−1	Ø，	+1，	−1	Ø，	+1，	−1
文字データ	A，	B，	C	D，	E，	F	G，	H，	I

2．ペンペン迷路

A．遊び方

プログラムを呼び出すときは，F2キーを押すか，または，

GAME　1 RETURN

とキーを押してください。ゲームが始まると，写真5のような画面になります。この画面に見えている縦・横の線が迷路です。ペンペンはこの上を動きまわることができます。

迷路の改造方法は，あとで説明します。

ゲームを始めるには，Iコントローラのボタンを押します。

ゲームが始まるとペンペンが出てきますから，Iコントローラの✣ボタンを押して迷路に沿って動かしてください。迷路には1～9の数字があります(写真6)。それを小さい順に1個通過するごとに，10点ずつの得点になります。全部取れるとパーフェクト‼

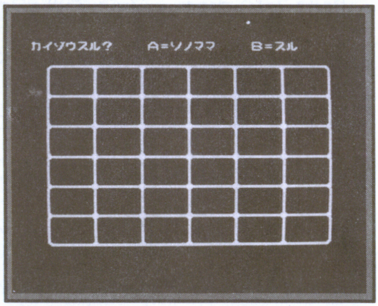


写真5

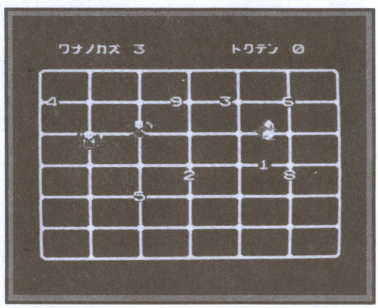


写真6

ところが，迷路にはカニさんがいて，ペンペンを追いかけて来ます。カニさんにつかまるとゲームは終わりです。つかまらないようにうまく逃げてください。ペンペンはワナを3つ持っていて，Aボタン

を押して迷路におくことができます。ワナにかかったカニさんは，しばらくの間動けなくなります。ゲームが終わると，「オシマイ」のメッセージが出ます。再びゲームを始めるには，Ⅰコントローラの START ボタンを押してください。それ以外のボタンを押すと「OK」が出て，プログラムは終了します。

B．改造の方法

ペンペン迷路ゲームは，BGグラフィックをいろいろと変えて遊べます。ゲームの始まりでBボタンを押すと，BGグラフィックモードに入ります。そこで迷路を変えてみましょう。

迷路には使えないキャラクタが2種類あります。ワナのキャラクタ（キャラクタコード表Bの219）と，BGグラフィックモードで点滅している「へ」（キャラクタコード表Bの94）です。

キャラクタだけではなく，迷路の形も変えることができます。写真7は迷路を変えた例です。このように自由にデザインすることができるのです。ただし，最初に出てきた迷路より外側には，はみ出さないようにしてください。はみ出すとエラーになることがあります。また，写真6のいちばん外側の迷路だけは，必ず作ってください。外側の迷路がないと，デザインした迷路が消えてしまったり，ペンペンが動けなくなってしまうことがあります。

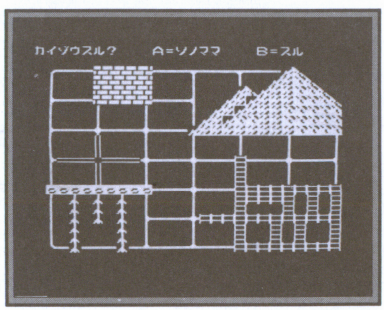


写真7

気に入ったBGグラフィックができたら，BGグラフィックモードで「FILE」を選ぶか，またはBASICのSAVE S命令を使って

テープにセーブしておけば，あとでその画面を使うことができます。テープにセーブしたBGグラフィックを使うときは，BGグラフィックモードで「FILE」を選ぶか，BASICのLOADS命令で本体にロードします。「FILE」を選べば，ロードした絵をすぐに見ることができます。

① 色を変えてみましょう

迷路の色やバックの色を変えたいときは，新しく1行追加します。

105 PALETB Ø，10，48，19，37

この数字でRUNすると，バックの色が緑，迷路の色がピンク，また，「オシマイ」のメッセージの枠が白と紫になります。この色を変えたいときは，図8の色コード表を見ながら，いろいろな数字にしてみましょう。色を変えるとゲームをやめても，その色のままになっています。元に戻したいときは，CTRキーを押しながらDキーを押してください（色表示はテレビの機種によって多少異なる場合があります）。

② ペンペンやカニさんをほかのアニメキャラクタに変えてみましょう

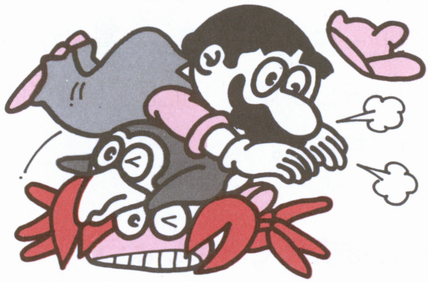
ペンペンを，ほかのアニメキャラクタに変えたいときは，39Ø行と58Ø行を変更してください。

```
39Ø DEF MOVE(7)=SPRITE (4,Ø, …以下略
58Ø DEF MOVE(7)=SPRITE (4,S, …以下略
```

4をØ～15の数字にすると，ペンペン以外のアニメキャラクタになります。どのアニメキャラクタになるのかは，本書の付録P.128（下段）をご覧ください。

たとえば，BGグラフィックで迷路をハシゴにしたときは，この数字をØにして，マリオを出すと自然です。

カニさんを変えたいときは，次の3つの行の14を同じように変更してください。



```
430 DEF MOVE (F)=SPRITE (14, 1, …
820 DEF MOVE (F)=SPRITE (14, 1…
1070 DEF MOVE (F)=SPRITE (14, …
```

③ 得点を変えてみましょう

このゲームでは，数字を1つ拾うと10点ですが，この得点を変えてみましょう。

```
630 T=T+10:LOCATE……以下略
```

得点は上のように，630行の10で決めています。この10を20や100にすると，数字を拾ったときの得点が変わります。あまり大きな数字にすると，エラーになることがあります。

④ ワナの数を変えてみましょう

ワナの数は360行で決めています。ワナの数を変えたいときは，360行の3をほかの数字にします。

```
360 PX=13*8+12:PY=2*8+16:C=3
:T=0:FW=0:F=0
```

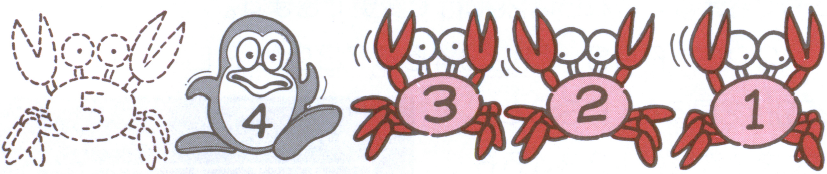
でも，あまりワナの数を多くすると，ゲームがやさしくなりますから，少なくするほうがいいかもしれません。

⑤ カニさんの数を変えてみましょう

ＫＮがカニさんの数を表しています。37Ø行にＫＮ＝2というのがありますから，この数字を変えてみましょう。

37Ø　ＫＮ＝2：Ｗ＝219

ただし，アニメキャラクタは，5つ以上が横に並ぶと，5つめは画面から消えてしまいます。カニさんの数は3にしておかないと，ペンペンが見えなくなってしまうことがあります。カニさんの数をØ



以下にすると，エラーになってしまいます。ですから，カニさんの数は1か2か3にしておいてください。

⑥ ワナをうまくおくことができないときは……

ワナをうまくおくことができなかったり，ワナを使った後に迷路がおかしくなったときは，555行をそのまま49Ø行にしてください。この方法は，まず，

LIST　555RETURN

で555行を画面に出します。

そしてカーソルを左端の5の上に動かして，行番号だけを

49Ø と打ち直し，RETURN キーを押します。

これで555行が490行にコピーされました。あとは，

　　5 5 5 RETURN

とキーを押して，555行を消してしまいます。

［ある行をほかの行に移したいときは，このようにします。自分でプログラムを作るときなどに便利ですから，この方法を覚えておくとよいでしょう。］

⑦　ペンペンが拾えるものを数字以外のキャラクタにしてみましょう

ペンペンは数字の１～９を，順番に拾えるようになっています。この順番や個数を変えたり，数字以外のアルファベットやカタカナにもできます。これらは290行で変更できます。

　290　PS$="123456789"：PL=……以下略

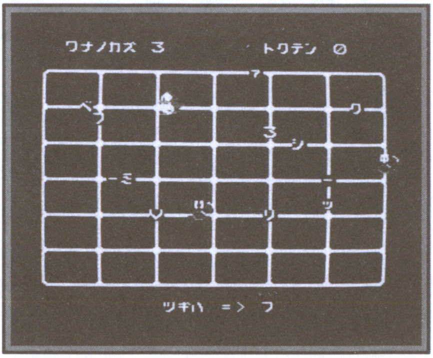


写真8

290行を見ると，PS$という文字変数に"123456789"を代入しています。この" "ではさまれた中身が拾えるキャラクタです。この中に書いた順番に，書いた個数だけを拾えるようにできます。たとえば，

　"98765"

　"ABCDEFGHIJKLMN"

　"MARIO"

　"ファミリーベーシックV3"　（写真8）

などにしておけば良いのです。ただし，" "の間に何も書かなか

ったり，書いた文字が31文字を越えると，エラーになりますから注意してください。また，数字はすべて１桁で，″１２１３２″のように，同じ文字が２回以上あってもかまいませんが、″１１２３２″のように，同じ文字を続けて書いてはいけません。

⑧ プレイ中に次に何を拾うかを画面に出してみましょう

新しく２行を追加すると，次に拾うキャラクタを，画面に出しておくことができます。こうするとゲームが進めやすくなります。

```
335 LOCATE 1Ø，22：PRINT ″ツギハ =
> ″；LEFT$（PS$，1）；
645 LOCATE 17，22：PRINT MID$（P
S$，PS，1）；
```

⑨ ボーナス得点をつけましょう

特別のキャラクタを拾うと，ボーナス得点がもらえるようにしてみましょう。

まず，29Ø行の″　″の中からボーナスキャラクタを１種類決めます。たとえば拾うものが，

″ＡＢＸＣ１２Ｘ３ＸＺ″



だとして，その中から「Ｘ」を，ボーナスキャラクタにしようと決

めたとします。290行の最後に，次のように付け足します。

```
290 PS$="ABXC12X3XZ" :PL=LEN(
                拾うもの
PS$):PS=1:B$="X"
          付け足す  ↑ボーナスキャラクタをかく
```

次に新しく605行を追加します。

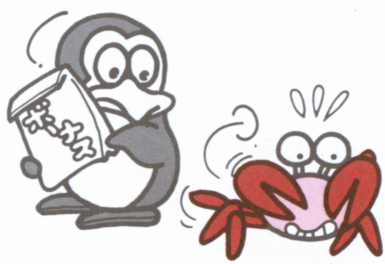
```
605 IF S$=B$ THEN T=T+100
                       ボーナス点
```

こうしておくと，ボーナスキャラクタ「X」を拾ったときにだけ，10点のほかに，ボーナス得点100点がもらえるようになります。ボーナス得点も，欲張ってあまり大きくしてはいけません。

得点の合計が，32767を越えるとエラーになります。

⑩ ボーナスキャラクタを拾うとカニさんが逃げるようにしましょう

ボーナスキャラクタを拾うと，カニさんが逃げて行くようにしてみましょう。ボーナスキャラクタの決め方は，ボーナス得点のときと同じで，290行に次のように付け足します。



```
290 PS$="12B34B43B21":PL=LEN(PS$)
            拾うもの
:PS=1:B$="B"
  付け足す  ↑ボーナスキャラクタ「B」
```

次に新しく606行と785行を追加します。

```
606 IF S$=B$ THEN DEF MOVE(6)
=SPRITE(11, Ø, 1, 255, Ø, Ø) :POSI
TION 6, 116, 1Ø:MOVE 6
785 IF MOVE(6)<>Ø THEN JX=-J
X:JY=-JY
```

このように改造すると，ボーナスキャラクタを拾ったときには，画面の上の方に，ニタニタが出てきて笑いだします。そして，今まで追いかけてきていたカニさんが，反対に逃げ始めます。ニタニタが笑っている間は，カニさんが逃げ続けます。ニタニタが笑うのをやめると，カニさんはまた追いかけてきます。ニタニタが笑っている時間は，6Ø6行のDEF MOVEで決まります。

```
606 IF S$=B$ THEN DEF MOVE(6)
=SPRITE(11, Ø, 1, 255, Ø, Ø)……以下略
```

この255を，もっと小さな数字に変えてみましょう。でも，あまり小さな数字にすると，カニさんが逃げるまもなく，ニタニタが笑うのをやめてしまいますから，小さくても5Øくらいにしておいたほうがよいでしょう。

⑪ タイマーを作ってみましょう

これまでは，ペンペンに有利な改造が多かったので，今度はタイマーを作って，ある時間が過ぎるとゲームが終わるようにしましょう。タイマーは，3行追加するだけで作ることができます。



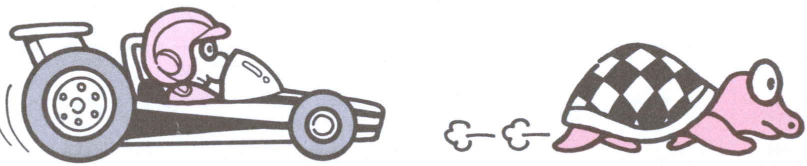
カメ

```
465 DEFMOVE(5)=SPRITE(13, 3, 4
5, 128, Ø, Ø):POSITION 5, 116, 21
Ø:MOVE 5
```

自動車

```
466 DEFMOVE(4)=SPRITE(6, 3, 25
, 128, Ø, Ø):POSITION 4, 16, 210:
MOVE 4
```

```
475 IF CRASH(5)=4 GOTO 112Ø
```



このタイマーは，自動車とカメさんの追いかけっこのタイマーです。この3行を追加してRUNすると，ゲームが始まったときに，画面の下の真ん中にカメさんが出てきて，右に向かって歩き始めます。そのカメさんを，左の方から自動車が追いかけてきます。自動車がカメさんに追い付くと，時間切れということになり，ゲームが終わります。

タイマーを作るときには注意が必要です。それは，拾うキャラクタの個数によって，どのくらいの時間で自動車がカメさんに追い付くかを，変えなければならないからです。たとえば，拾うキャラクタが3つしかないのに，時間がたっぷりあったのでは，少しもおもしろくありません。拾うキャラクタの個数によって，465行と466行の数字を変えてください。（この数字が速さを決めています。）

```
465 DEFMOVE(5)=SPRITE(13,3,45,
128, Ø, Ø) :………以下略
466 DEFMOVE(4)=SPRITE(6, 3, 25,
```

128，Ø，Ø）：………以下略

この例では，拾うキャラクタが12個くらいで，ちょうどよくなる速さです。拾うキャラクタがもっと多ければ，両方の数字も少し大きくし，少なければ小さくします。速さに応じて，何回か数字を変えながら，ゲームをしてください。数字を変えるときは，カメさんの数字が自動車の数字の，1.8倍くらいになるようにしてください。この例では，

自動車の数字　X　1.8　＝　カメさんの数字

25　X　1.8　＝　45

になっています。こうしておけば，画面の右端で，自動車がカメさんに追い付きます。

⑫　カニさんの動きを変えてみましょう。

迷路が，写真9のように切れているところでは，ペンペンは通れないのに，カニさんだけは平気で通っていきます。カニさんも通れないようにするには，次のように変えます。

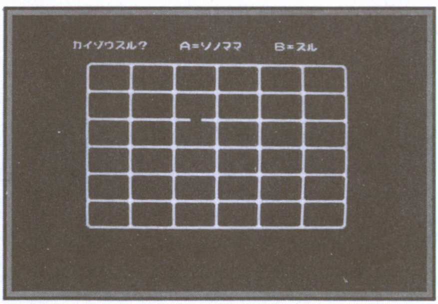
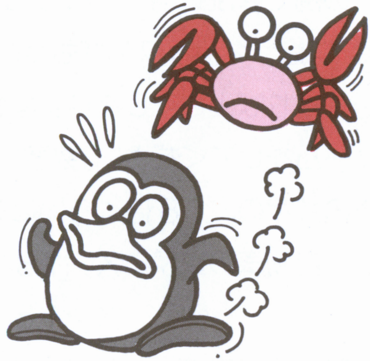


写真9

```
880　IF（KD=3　AND　JX=1）OR（KD=7
　AND　JX=-1）OR（KD=5　AND　JY=1）
OR（KD=1　AND　JY=-1）　THEN　1Ø6Ø
```

変更するのは，88Ø行の数字1ヵ所だけです。「1Ø6Ø」を，「94Ø」にしてください。これで，写真9のように迷路が切れているところは，カニさんも通れなくなります。

⑬ カニさんが追いかけてくる速さを変えてみましょう



カニさんが，ペンペンを追いかけてくる速さは，あまり速くはありませんので，ペンペンは簡単に数字を全部拾うことができます。そこで，プログラムを少し変えると，とても難しいゲームになります。こうするには，1行を変更してから，別に新しく1行を追加します。

〈変更する〉

```
700  F=F+1:IF F>KN-1 THEN F=Ø
700  FOR F=Ø TO KN-1
```

〈追加する〉

```
1105 NEXT
```

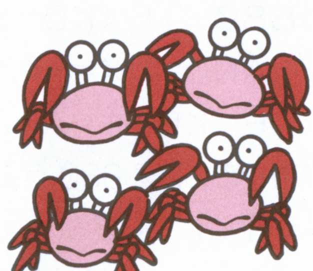
こうすると動く順番が，

ペンペン→カニ1→ペンペン→カニ2→ペンペン→カニ1……

だったのが，

ペンペン→カニ1→カニ2→ペンペン→カニ1→カニ2

の順番になり，カニさんが速く追いかけてくるようになります。



この改造をすると，ゲームが難しくなりますから，ワナの数を増やしたり，ボーナスキャラクタを，決めておいたほうがよいでしょう。

もし，カニさんの数を3以上にすると，難しくてゲームになりません。

また，ペンペンの動きがぎこちなくなりますので，気になる人は，580行を次のように変えてください。

580 DEF MOVE(7)=SPRITE (4, S, 2, 4, Ø, Ø)：……以下略

「2」を「3」に変えると，少しなめらかになります。





ペンペン迷路プログラムリスト

```
100 CLEAR:CGEN 2:CGSET 1,1:S
CREEN 0,1
110 SPRITE ON:PLAY "V15T1:V1
5T1"
```

●ゲームの準備

```
120 IF SCR$(1,2)<>" " GOTO 2
40
```

●すでにBGが描いてあれば240行に行く

```
130 '■メイロ
140 FOR JX=1 TO 21 STEP 4:FO
R JY=2 TO 20 STEP 3
150 LOCATE JX,JY:PRINT "┼───
┼":IF JY>19 GOTO 180
160 LOCATE JX,JY+1:PRINT "│
 │"
170 LOCATE JX,JY+2:PRINT "│
 │"
180 NEXT:NEXT
190 FOR S=5 TO 17 STEP 3:LOC
ATE 1,S:PRINT "├":LOCATE 25,
S:PRINT "┤":NEXT
200 LOCATE 1,2:PRINT "┌";
210 FOR S=0 TO 4:PRINT "───┬
";:NEXT:PRINT "───┐";
220 LOCATE 1,20:PRINT "└";
230 FOR S=0 TO 4:PRINT "───┴
";:NEXT:PRINT "───┘"
```

●迷路を描く

```
240 '■トルモノ
250 SCREEN 0,0:VIEW:PRINT "カ
イゾ゛ウスル?    A=ソノママ    B=スル"
260 K=STRIG(0):IF K=0 GOTO 2
60
270 IF K=4 THEN BGTOOL
```

●改造するときは，BG グラフィックモードにする

```
280 PLAY "O3C1DE:O3E1FG"
290 PS$="123456789":PL=LEN(P
S$):PS=1
300 FOR S=1 TO PL
310  SX=2+RND(24):SY=2+RND(1
8):S$=SCR$(SX,SY)
320  IF S$=" " OR INSTR(PS$,
S$)>0 GOTO 310
330 LOCATE SX,SY:PRINT MID$(
PS$,S,1)
340 NEXT
```

●拾うものをおく

```
350 '■ペンスウ
360 PX=13*8+12:PY=2*8+16:C=3
```

※本書リストに出てくる「＊」は，「＊」と同じ文字です。画面に出てくる文字と，多少異なりますので，ご了承ください。

```
:T=0:FW=0:F=0
370 KN=2:W=219:KX=0:KY=0:KD=
0
380 LOCATE 0,0:PRINT "   ﾜﾅﾉ
ｶｽﾞ";C;"        ﾄｸﾃﾝ";T
390 DEF MOVE(7)=SPRITE (4,0,
2,4,0,0):POSITION 7,PX,PY:MO
VE 7
400 DIM E(KN-1)
410 '■ｶﾆﾃﾞﾙ
420 FOR F=0 TO KN-1
430 DEF MOVE(F)=SPRITE(14,1,
2,4,0,3)
440 SX=1+RND(26):SY=10+RND(1
0):IF SCR$(SX,SY)=" " OR SCR
$(SX,SY-1)=" " GOTO 440
450 POSITION F,SX*8+12,SY*8+
24:MOVE F
460 NEXT
470 '■■■■ﾒｲﾝ■■■■
480 '■■■■■ﾍﾟﾝｷﾞﾝ
500 '■ｷｰｲﾝ/ｳｺﾞｸ
510 S=0:SX=0:SY=0:K=STICK(0)
:IF K=0 GOTO 590
520 IF K=1 THEN S=3:SX=1
530 IF K=2 THEN S=7:SX=-1
540 IF K=4 THEN S=5:SY=1
550 IF K=8 THEN S=1:SY=-1
555 PX=(XPOS(7)-12)/8:PY=(YP
OS(7)-16)/8
560 S$=SCR$(PX+SX,PY+SY)
570 IF S=0 OR S$=" " GOTO 59
0
580 DEF MOVE(7)=SPRITE (4,S,
2,4,0,0):POSITION 7,PX*8+12,
PY*8+16:MOVE 7
590 '■ｽｳｼﾞﾋﾛｳ
600 IF S$<>MID$(PS$,PS,1) GO
TO 650
610 PLAY "O4C0#CD#DEF#FG#G"
620 LOCATE PX+SX,PY+SY:PRINT
 CHR$(199);
630 T=T+10:LOCATE 21,0:PRINT
 T
640 PS=PS+1:IF PS>PL THEN PL
AY "O4C3DEFG:O4E3FGAB":GOTO
1120
650 '■ﾜﾅｵｸ
```

●変数を定義してペンペンを出す

●カニさんを出す

●✚ボタンが押されたら，ペンペンをその方向に動かす

●数字を拾ったら，スコアに10点を加える

```
660 IF STRIG(0)<>8 OR INSTR(
PS$,S$)<>0 OR C<1 OR FW=1 GO
TO 690
670 PLAY "O2E2C":W$=S$:LOCAT
E PX+SX,PY+SY:PRINT CHR$(W)
680 FW=1:C=C-1:LOCATE 8,0:PR
INT C
690 '■■■■■カニ
700 F=F+1:IF F>KN-1 THEN F=0

710 '■シビレテル?
720 KD=VCT(F):IF KD>0 GOTO 7
60
730 IF MOVE(F)<>0 THEN PLAY
"O0C0EGCEG":GOTO 1080
740 '■シビレオワリ
750 KD=E(F)
760 '■ザヒョウ
770 KX=(XPOS(F)-12)/8:JX=SGN
(PX-KX)
780 KY=(YPOS(F)-16)/8:JY=SGN
(PY-KY)
790 SK$=SCR$(KX,KY):SR$=SCR$
(KX+1,KY):SL$=SCR$(KX-1,KY):
SD$=SCR$(KX,KY+1):SU$=SCR$(K
X,KY-1)
800 '■ワナ?
810 IF SK$<>CHR$(W) GOTO 850

820 DEF MOVE(F)=SPRITE (14,0
,1,200,1,2):POSITION F,KX*8+
12,KY*8+16:MOVE F
830 E(F)=KD:LOCATE KX,KY:PRI
NT W$:FW=0
840 GOTO 1080
850 '■■■オイカケル
860 IF(SR$+SL$<>"  " AND SD$
+SU$<>"  ") GOTO 910
870 '■■チョクシン
880 IF(KD=3 AND JX=1)OR(KD=7
 AND JX=-1)OR(KD=5 AND JY=1)
OR(KD=1 AND JY=-1) GOTO 1060

890 '■ハンテン
900 KD=(KD+4)+8*(KD>3):GOTO
1060
910 '■■ブンキ
920 '■イケル
```

● Aボタンが押されたらワナをおく (660–680)

● 2匹のカニさんのどちらを動かすかを決める (700)

● カニさんがワナでシビレていればそのまま，シビレ終わっていたらワナからぬけ出す (710–750)

● カニさんのスプライト座標をキャラクタ座標に変えて，ペンペンとの位置関係を計算する (760–790)

● カニさんがワナにかかったらシビレさせる (800–840)

● カニさんがいるところが分岐点かどうかを判定する (850–860)

● そのまま進む (870–880)

● 反転させる (890–920)

```
930 IF JX*JY<>0 AND RND(2)=1
 GOTO 960
940 IF JX=1 AND SR$<>" " THE
N KD=3:GOTO 1060
950 IF JX=-1 AND SL$<>" " TH
EN KD=7:GOTO 1060
960 IF JY=1 AND SD$<>" " THE
N KD=5:GOTO 1060
970 IF JY=-1 AND SU$<>" " TH
EN KD=1:GOTO 1060
980 '■ｲｹﾅｲ
990 S=RND(4)+1
1000 ON S GOTO 1010,1020,103
0,1040
1010 IF SR$<>" " THEN KD=3 G
OTO 1060
1020 IF SL$<>" " THEN KD=7 G
OTO 1060
1030 IF SD$<>" " THEN KD=5 G
OTO 1060
1040 IF SU$<>" " THEN KD=1 G
OTO 1060
1050 GOTO 1010
1060 '■ｳｺﾞｸ
1070 DEF MOVE(F)=SPRITE (14,
KD,1,4,0,3):POSITION F,KX*8+
12,KY*8+16:MOVE F
1080 '■ﾌﾞﾂｶｯﾀ?
1090 IF CRASH(7)>=0 THEN PLA
Y "O1B2AGFEDC":GOTO 1120
1100 '■■
1110 GOTO 470
1120 '■■■ｵﾜﾘ
1130 FOR JX=6 TO 8:READ S$:L
OCATE 9,JX:PRINT S$:NEXT
1140 SPRITE OFF
1150 DATA "■■■■■■■■■■"
1160 DATA "■■ ｵｼﾏｲ ■■"
1170 DATA "■■■■■■■■■■"
1180 S=STRIG(0):IF S=0 GOTO
1180
1190 IF S=1 THEN RUN
1200 END
```

- 分岐点でペンペンに近付く方向をさがす（930〜970）
- 分岐点でペンペンに近付く方向に進めなかったら，ほかに進める方向をさがす（990〜1050）
- いろいろ判定して決めた方向にカニさんを進ませる（1060〜1070）
- ペンペンがカニさんにつかまったらゲームオーバーにする（1080〜1090）
- → メインルーチンの最初に戻る（1110）
- ゲームオーバーの表示をする（1120〜1170）
- STARTボタンが押されたら，またゲームを始める。それ以外のボタンでプログラムを終わる（1180〜1200）

ペンペン迷路変数リスト

PS$	拾うものの文字列
PL	拾うものの数
PS	いくつ目まで拾ったか
C	ワナの数
T	スコア
FW	ワナを置いたかどうかのフラグ
W	ワナの BG キャラクタコード
F	どちらのカニさんを動かすかを決めるフラグ
KN	カニさんの数
KX, KY	カニさんの座標
KD	カニさんの進む方向
E(　)	カニさんの状態を示すフラグ
SR$, SL$, SU$, SD$, SK$	カニさんのまわりの迷路を知るための変数

3. マリオワールド

A. 遊び方

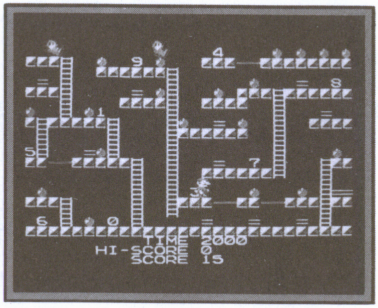


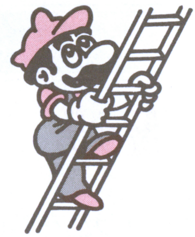
写真10

プログラムを呼び出すときは，F3キーを押すか，または

GAME 2，1 RETURN

と押してください。このゲームには，BGグラフィックが付いています。1人用ゲームですから，Iコントローラを使います。ゲームが始まったときは，写真10のような画面になっています。床の上には0〜9の10個の数字と10個のリンゴがあります。コントローラの✚ボタンを使ってマリオを左右に歩かせたり，ハシゴを登り降りさせて，数字とリンゴを取ってください。数字は1個30点，リンゴは1個15点です。数字は0，1，2……9の順に取らないと，得点にはなりません。マリオは，床の端やハシゴから飛び降りることができます。コントローラのAボタンを押すと，マリオが真上にジャンプし，✚ボタンの左か右を一緒に押すと，その方向にジャンプします。床の上に「＝」がありますが，これはジャンプ台です。マリオがこの上に立ち止まると，自動的にひとつ上の床に飛び乗ることができます。数字を取るときや，ジャンプ台を使うときは，必ず立ち止まるようにしてください。立ち止まらないと，判定されません。

Aボタンを使ってうまくジャンプすると，ニタニタを飛び越すこともできます。ニタニタは画面の左右に消えると，またすぐ上の方



から出てきます。

ハシゴを登りたいときは，ハシゴの真下で立ち止まってから，✜ボタンを上に押します。また，Aボタンを使いハシゴに向かってジャンプすると，すばやく登ることができます。

ゲームオーバーは，画面の下のＴＩＭＥがØになるか，ニタニタにつかまったときです。

もう１度ゲームをしたいときは，[Ｉ]コントローラのＳＴＡＲＴボタンを押すと，ゲームが始まります。ゲームをやめたいときは，[STOP]キーを押してプログラムを止めてください。

Ｂ．改造の方法

①　ＢＧグラフィックを変えてみましょう

マリオワールドは，ＢＧグラフィックを変えて，自分の好きな画面にすることができます。[STOP]キーでプログラムを止めてから

[B][G][T][O][O][L][RETURN]



とキーを押して，ＢＧグラフィックモードにしてください。ハシゴや床の場所を変えたり，リンゴの数を増やしたりしてみましょう。ただし，床やハシゴなどは決まったＢＧキャラクタを使ってください。それ以外のＢＧキャラクタは判定されません。数字はØ〜9がそれぞれ１つずつあるようにしておきましょう。

(注) BGTOOL で BG グラフィックを改造してから，ゲームを始

めるときは，F3キーではなく，必ずF8キーを押してください。そうしないと，改造したBGグラフィックは消えてしまいます。
気に入ったBGグラフィックができたら，BGグラフィックモードの「FILE」か，BASICのSAVES命令を使って，テープにセーブしておきましょう。テープからロードするときは，BGグラフィックモードの「FILE」か，BASICのLOADS命令を使います。
内蔵されたマリオワールドのプログラムと，テープにセーブしてあるBGグラフィックを組み合わせて遊ぶときは，まずF3キーを押すか，

GAME 2 RETURN

とキーを押して，プログラムを呼び出し，動き始めてから STOP キーでゲームを止めてください。次にテープからBGグラフィックをロードして，RUNしてください。なお，詳しくは，P.17 の『2-3内蔵プログラムの使い方』を参照してください。

② BGグラフィックの色を変えてみましょう

リストの40行にPALETB命令があります。その後に新しく3つの行を追加します。

40 PALETB 3，13，48，54，22：…以下略
（22：リンゴ）

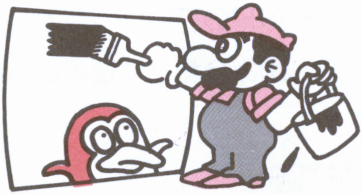
41 PALETB 2，13，48，54，25
（25：リンゴ）

42 PALETB 1，13，48，54，48
（48：床と床の間のバー）

```
43 PALETB Ø，4，48，54，39
```

バック　文字とハシゴ　床とジャンプ台

それぞれの色を自由に変えることができます。好きな色にしたいときは，色コード表（P．32）に従って変更してください。文字とハシゴの色を黒にすると，後でリストを出したときに，見えなくなってしまいますから注意しましょう。もしそうなってしまったときは，CTRキーを押しながらDキーを押せば，元の色(白)に戻ります。



③ アニメキャラクタを変えてみましょう

28Ø行の数字を変えると，ニタニタがほかのアニメキャラクタに変わります。

```
28Ø DEF MOVE（1）=SPRITE（11，D，…
```

11はニタニタのことですから，Ø～15のほかの数字にすればよいのです。数字をいくつにすると，どのアニメキャラクタになるのかは，本書の付録P.128（下段）をご覧ください。

54Ø行と63Ø行の数字を変更すると，ハシゴの登り降りや左右に動くときのアニメキャラクタが変わります。

```
54Ø DEF MOVE（7）=SPRITE（Ø，V，1…
63Ø IF K<4 THEN DEF MOVE（7）=
SPRITE（Ø，V，4－K，……
```

以下略

どちらの行もØのマリオになっていますが，たとえば54Ø行は4

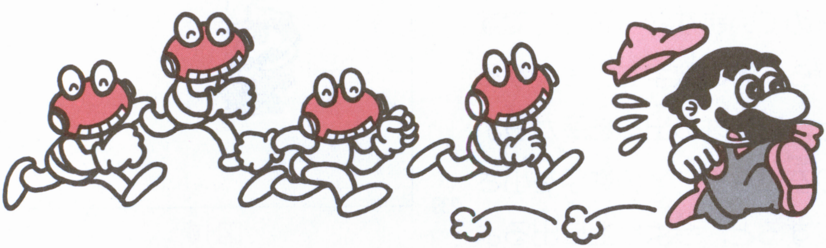
に，63Ø行は13に，というように変えます。

④ ニタニタの数を変えてみましょう

ニタニタは2匹で追いかけてきますが，たった1ヵ所変えるだけで，増やしたり減らしたりすることができます。変更するのは14Ø行で，

```
140 T=800:W=0:NN=1:X=116……以下略
```

この「1」を，ほかの数字にすればよいのです。ただし，この数字はニタニタの数より，1だけ小さい数にしなければいけません。今はニタニタが2匹なので，14Ø行の数字は，2より1少ない数字になっているわけです。4匹（数字は3）以上にすると，マリオがときどき見えなくなります。



⑤ 得点を変えてみましょう

数字は1個30点，リンゴは1個15点となっていますが，この得点は変えることができます。たとえば，数字を1個100点にするときは，

```
420 IF ASC(SCR$(XL,YL-1))=Z
THEN LOCATE XL,YL-1:PRINT "
":Z=Z+1:W=W+30:……以下略
```

の3Øを1ØØに変えてください。

リンゴの得点を1個50点にするときは，

880 LOCATE XB，YL－1：PRINT ″ ″
：R＝R＋1：W＝W＋15……以下略

15を50に変えてください。数字とリンゴの得点の合計が，32767を越えるとエラーになります。

⑥ マリオが最初に出てくる場所を変えてみましょう

マリオが画面に出てくる場所は140行のXとYで決めています。

140 T＝800：W＝0：NN＝1：X＝116：Y＝127：POSITION……以下略

Xは，画面の左端からの位置，Yは，画面の上からの位置を決めています（図10）。ですから，Xの数字を小さくすると，マリオの出る場所が今より左になり，Yの数字を小さくすると，今より上に出るようになります。この数字を変えるときは，Xが16～224，Yが16～168くらいにしておきましょう。BGグラフィックで自分の画面を作った人は，その画面に合わせて，マリオを出す場所を変えた方がよいでしょう。

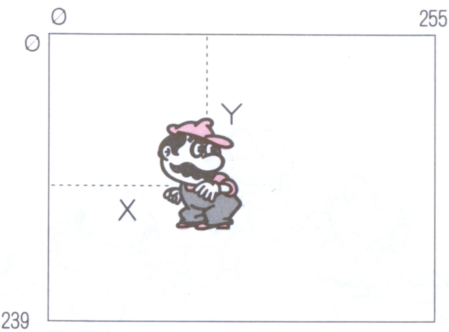


図10

⑦ タイマーを付けてゲーム時間を変えてみましょう

画面の下にTIMEという表示があります。ここの数字が800から10ずつ減っていって，0でゲームオーバーになっていますが，

８００という数字を変えることによって長くしたり短くしたりもできます。リストの最初のところに，

```
140 T=800:W=0:………以下略
```

ゲーム時間を決める数字がありますから，その数字を変えてください。ただし，ＢＧグラフィックで，リンゴの数が１５個以内で時間だけを長くすると，数字とリンゴを全部取った後，時間があり余って，ただマリオを動かすだけになってしまいます。リンゴを16個以上にしておけば，数字とリンゴを全部取ると，また新しく出てきますから，時間を長くしてもゲームを続けることができます。ですから，元のＢＧグラフィック画面を使って遊ぶときは，時間を短くした方がよいでしょう。

⑧　ジャンプ台でのジャンプの高さを変えてみましょう

マリオは，ジャンプ台を使ってひとつ上の床に飛び乗ることができます。ジャンプ台でのジャンプの高さを変えると，おもしろくなります。４１０行で，

```
410 IF SCR$(XL, YL-1)=K$ THEN
  G=-15:GOTO 770
```

Ｇ＝－１５の数字が，ジャンプの高さを決めています。
この数字を－２０に変えると，高くジャンプするようになります。ただし，ＢＧグラフィックの作り方によって，ものすごく高く飛んだり，すぐ上の床に飛び乗るだけだったりします。ＢＧグラフィックを作るときには，画面のあまり上のほうに，ジャンプ台を置かないようにしましょう。ジャンプしたときに，アニメキャラクタが画面からはみ出てエラーになり，プログラムが終わってしまうことが

あります。

次は，逆に低くしてみましょう。G＝−15の数字を，−10にすると，もう上の床に飛び上がれなくなってしまいます。

さて，最後に少し愉快なプログラムにしましょう。G＝−3にしてください。これでマリオがジャンプ台のところで立ち止まると，ダンシングマリオになってしまいます。



今度は，G＝0にしてRUNしてみてください。ジャンプ台が落とし穴に変わってしまいます。ジャンプ台のところでマリオが立ち止まると，ひとつ下の床まで落ちます。これはゲームのルールが変わったことにすると，また楽しく遊べます。落とし穴にしたときは，いちばん下の床から落ちると，ゲームオーバーになります。

⑨　Aボタンでのジャンプ幅を変えてみましょう

ニタニタが近づいてきたときに，Aボタンを押してうまくジャンプすると，マリオはニタニタを，飛び越すことができますが，改造すると，飛び越せなくなったり，もっと楽に飛び越すことができるようになります。



```
770 F=1:SPRITE S,X,Y:ERA 7::
SPRITE 3:POSITION 7,X,Y:S2=S
:K=K*((S=2)……以下略
```

リストの77Ø行に，「K＝K＊」というところがありますが，ここを変更すると，ジャンプ幅を変えることができます。たとえば，

K＝（K－1）＊

にすると，ぎりぎりのところでニタニタを飛び越せなくなります。

K＝（K－3）＊

にすると，✢ボタンを押しながらジャンプしても，ほとんど前へ進まなくなってしまいます。

K＝（K＋2）＊

のようにすると，ニタニタの飛び越しが，少し楽になります。

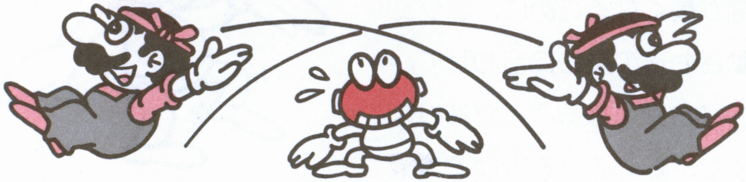
ゲームとしては使えませんが，たとえば，スーパージャンプにしたいときは，

K＝（K＋1Ø）＊

にすると，ニタニタを追いかけて，飛び越すこともできるようになります。それから，

K＝（K－2Ø）＊

にすると，マリオは後ろに飛んでしまいます。



⑩ Aボタンでのジャンプ力を変えてみましょう

⑨では，マリオがジャンプしたときに，どのくらい遠くまで飛べるのか，いろいろ改造してみました。そこで今度は，どのくらい高くジャンプできるのか，Aボタンのジャンプ力を，いろいろ変えて

みましょう。

改造する行は，760行です。

760　G=－9：X=X+……以下略

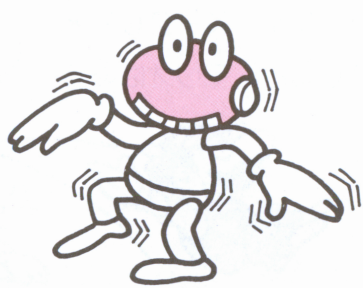
760行のはじめのところに，「G=－9」というのがありますが，この「－9」という数字を,「－15」や「－20」のような，マイナスのついた大きな数字にすると，より高くジャンプできるようになります。

ところが，あまり上の方からジャンプすると，画面の外に飛び出してしまい,「OK」の文字が表示され，プログラムが止まってしまいます。そこで，次の1行を新しく追加してください。

1025　IF　Y<16　THEN　Y=16：YL=2：RESUME

この1行を追加しておけば，画面の上の方でジャンプしても，プログラムの実行が，中止されることはありません。

なお，ジャンプする高さが高くなるほど，1回のジャンプにかかる時間が長くなるために，ニタニタの動きもその長さに合わせて，少しぎこちなくなることがあります。また，画面の左右両端でジャンプすると，マリオが空中で止まってしまい，キーを押しても動かなくなってしまうことがあります。もし，止まってしまったら，STOPキーを押して，1度プログラムを中止させ，RUNし直してください。



ジャンプする高さを高くしすぎると，たとえば「－25」や「－

3Ø」という大きな数字にすると，ジャンプした後，床を通り越して，落ちてしまうことがあります。ですから，数字は「－2Ø」くらいまでにしておいた方がよいでしょう。





ジャンプの高さを変えたときは，写真11のように，BGTOOL命令を使って，ハシゴや床を少し消しておきます。そして，広い場所を作っておいて，そこでジャンプすると，改造の効果をよく見ることができますから，試してみてください。

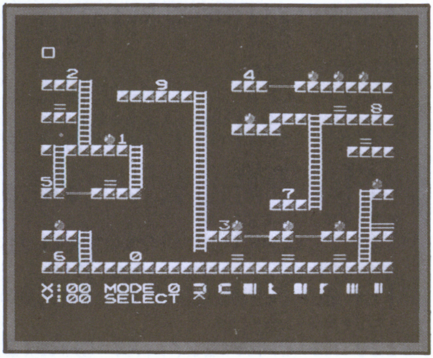


写真11

⑪ マリオの歩き始めをすばやくしましょう

マリオを左右に動かそうとするとき，✚ボタンの左か右を押すと，少しの間ゆっくりと歩いてから，速く歩きだしています。これは，動きが自然に見えるような工夫なのですが，すばやく動かしたい人は，63Ø行を変えてください。

```
630 IF K<4 THEN DEF MOVE(7)=
SPRITE(Ø,V,4-K,255):……以下略
```

（　）の中の4－Kというところを，数字の1にします。すると，マリオはいきなり速く歩き始めるようになります。

マリオワールドプログラムリスト

●準備

```
10 CGSET 1,0
20 ON ERROR GOTO 910
30 W=F=K=T=R=Z=NI=NN=XE=YE=X
N=YN=XB=XL=YL=S2=S1=S=G=V=X=
Y=I=D:H=0:S$=G$
40 PALETB 3,13,48,54,22:SPRI
TE ON
```

●ジャンプしているときのマリオのパターンを決める

```
50 K$=CHR$(24):R$=CHR$(25):H
$=CHR$(26):U$=CHR$(27)
60 DEF SPRITE 0,(0,1,0,1,0)=
R$+K$+U$+H$
70 K$=CHR$(12):R$=CHR$(13):H
$=CHR$(14):U$=CHR$(15)
80 DEF SPRITE 1,(0,1,0,1,0)=
R$+K$+H$+U$
90 DEF SPRITE 2,(0,1,0,0,0)=
K$+R$+H$+U$
100 K$=CHR$(20):R$=CHR$(21):
H$=CHR$(22):U$=CHR$(23)
110 DEF SPRITE 3,(0,1,0,0,0)
=K$+R$+H$+U$
```

●準備

```
120 K$=CHR$(227):R$=CHR$(215
):H$=CHR$(210):U$=CHR$(192):
DIM Q(3),D(3)
130 'ｼｮｷ ｾｯﾃｲ
140 T=800:W=0:NN=1:X=116:Y=1
27:POSITION 7,X,Y:GOSUB 890:
V=0:GOTO 540
```

●時間とスコアの判定

```
150 'ﾀｲﾑ ﾄ ﾄｸﾃﾝ ﾉ ﾊﾝﾃｲ
160 T=T-1:IF T MOD 10=0 THEN
 LOCATE 14,21:PRINT T;"  ":I
F T=0 THEN 660
170 IF Z=58 THEN IF R>15 THE
N GOSUB 890
```

```
180 'ﾆﾀﾆﾀ ﾉ ﾌﾟﾛｸﾞﾗﾑ
190 I=-(I+1)*(I<>NN)
200 XN=XPOS(I):YN=YPOS(I)
210 IF XN<12 OR XN>222 OR YN
>188 THEN XN=RND(27)*8+12:YN
=7:POSITION I,XN,YN:D=0:Q(I)
=5:GOTO 280
220 XE=(XN-8)/8:YE=(YN-2)/8
230 G$=SCR$(XE,YE)
```


※本書リストに出てくる「⁎」は，「＊」と同じ文字です。画面に出てくる文字と，多少異なりますので，ご了承ください。

```
240 D=Q(I)
250 IF G$<>U$ THEN IF YL<>YE
 OR G$<>H$ THEN D=5:Q(I)=3-4
*(X<XN):GOTO 270
260 IF D(I)=D THEN 300
270 XN=XE*8+12:YN=YE*8+7:POS
ITION I,XN,YN
280 DEF MOVE(I)=SPRITE(11,D,
2,255+251*(D=5)):MOVE I:D(I)
=D
290 'ﾌﾞﾂｶﾘ ﾉ ﾊﾝﾃｲ
300 NI=CRASH(7):IF NI>-1 THE
N IF ABS(YPOS(NI)-YPOS(7))<1
4 AND ABS(XPOS(NI)-XPOS(7))<
14 THEN 660
310 IF F=1 THEN 780
320 'ﾏﾘｵ ﾉ ﾌﾟﾛｸﾞﾗﾑ
330 S1=S:S=STICK(0):IF S>2 T
HEN S=3-(S>7)
340 X=XPOS(7):Y=YPOS(7)
350 XL=X/8-1:YL=(Y-2)/8
360 ON S GOTO 550,550,480,51
0
370 '
380 V=0:K=0:CUT 7
390 IF SCR$(XL,YL)=" " THEN
IF SCR$(XL,YL-1)<>H$ THEN GO
SUB 870:GOTO 750
400 XB=XL+(S1=1)-(S1=2):IF S
CR$(XB,YL-1)=R$ THEN GOSUB 8
80
410 IF SCR$(XL,YL-1)=K$ THEN
 G=-15:GOTO 770
420 IF ASC(SCR$(XL,YL-1))=Z
THEN LOCATE XL,YL-1:PRINT "
":Z=Z+1:W=W+30:LOCATE 14,23:
PRINT W;
430 IF STRIG(0)<>8 THEN 470
440 IF SCR$(XL-1,YL)=U$ THEN
 760
450 IF SCR$(XL,YL)=U$ THEN 7
60
460 IF SCR$(XL+1,YL)=U$ THEN
 760
470 GOTO 160
480 V=5:G$=SCR$(XL,YL)
490 IF G$<>H$ THEN IF G$<>U$
 THEN S=0:V=0:GOTO 750
```

●ニタニタがマリオを追(お)いかける

●マリオがニタニタにつかまった判定(はんてい)

●Ⅰコントローラの✜ボタンが押(お)された判定(はんてい)

●数字(すうじ)をひろったら30点(てん)

●マリオの足元(あしもと)の床(ゆか)の判定(はんてい)

```
500 GOTO 520
510 V=1:G$=SCR$(XL,YL-1)
520 IF G$<>H$ THEN S=0:GOTO
380
530 IF S1=0 OR K>0 THEN GOSU
B 870:K=0
540 DEF MOVE(7)=SPRITE(0,V,1
,4):MOVE 7:SPRITE S2:PAUSE 5
:GOTO 160
```

●マリオがハシゴを登り降りする

```
550 V=4*S-1:K=1-(K+(K>3))*(S
=S1)
560 G$=SCR$(XL+1+2*(S=2),YL)
:S$=SCR$(XL,YL)
570 XB=XL+1+2*(S=1):IF K>2 T
HEN IF SCR$(XB,YL-1)=R$ THEN
 GOSUB 880
580 IF SCR$(XL,YL-1)=R$ THEN
 XB=XL:GOSUB 880
590 IF G$<>U$ AND G$<>H$ THE
N X=X+6*(S$<>" ")*((S=1)-(S=
2)):GOTO 750
600 IF S$=" " THEN S=S1:GOTO
 750
610 IF STRIG(0)=8 THEN IF G$
=U$ OR S$=U$ OR SCR$(XB,YL)=
U$ THEN 760
620 NI=CRASH(7):IF NI>-1 THE
N IF ABS(XPOS(NI)-XPOS(7))<1
4 THEN 660
630 IF K<4 THEN DEF MOVE(7)=
SPRITE(0,V,4-K,255):MOVE 7:S
PRITE S2
640 GOTO 160
```

●マリオを左右に歩かせて
リンゴや床の判定をする

```
650 'ｵﾜﾘ ﾉ ﾌﾟﾛｸﾞﾗﾑ
660 SPRITE S:SPRITE S1:SPRIT
E S2:ERA 7
670 FOR I=0 TO NN:ERA I:POSI
TION I,0,0:D(I)=0:NEXT
680 LOCATE 8,9:PRINT"*******
******"
690 LOCATE 8,10:PRINT"* GAME
 OVER *"
700 LOCATE 8,11:PRINT"******
*******"
710 IF W>H THEN H=W:LOCATE 1
4,22:PRINT H;
720 IF STRIG(0)=1 THEN 140
730 GOTO 720
```

●ゲームオーバーを表示して
再ゲームの入力待ち

```
740 ´ｼﾞｬﾝﾌﾟ ﾉ ﾌﾟﾛｸﾞﾗﾑ
750 G=2:Y=Y+3:GOTO 770
760 G=-9:X=X+(2*K+6)*((S=2)-
(S=1)):Y=Y-12
770 F=1:SPRITE S,X,Y:ERA 7::
SPRITE 3:POSITION 7,X,Y:S2=S
:K=K*((S=2)-(S=1)):GOTO 160
780 NI=CRASH(7):IF NI>-1 THE
N IF ABS(YPOS(NI)-Y<12 THEN
660
790 G=G+3+3*(G>6):X=X+K:Y=Y+
G:SPRITE S,X,Y:POSITION 7,X,
Y
800 XL=X/8-1:YL=(Y-2)/8:G$=S
CR$(XL,YL):S$=SCR$(XL,YL-1)
810 IF G$=U$ THEN GOSUB 860:
K=1:GOTO 540
820 IF S$=H$ THEN GOSUB 860:
SPRITE 3,X,Y:SPRITE S2:S2=3:
GOTO 160
830 IF G$=H$ THEN GOSUB 860:
V=0:GOTO 540
840 GOTO 780
850 ´SUB ﾉ ﾌﾟﾛｸﾞﾗﾑ
860 GOSUB 870:F=0:K=0:SPRITE
 S,X,Y:RETURN
870 CUT 7:X=XL*8+12:Y=YL*8+7
:POSITION 7,X,Y:RETURN
880 LOCATE XB,YL-1:PRINT " "
:R=R+1:W=W+15:LOCATE 14,23:P
RINT W;:RETURN
890 VIEW:I=0:F=0:K=0:C=0:R=0
:Z=48:LOCATE 10,21:PRINT"TIM
E";T:LOCATE 6,22:PRINT"HI-SC
ORE";H:LOCATE 9,23:PRINT"SCO
RE";W;
900 RETURN
910 ´ｴﾗｰ ｼｮﾘ ﾉ ﾌﾟﾛｸﾞﾗﾑ
920 IF ERL=560 THEN XL=-27*(
XL<1):GOSUB 870:MOVE 7:RESUM
E 630
930 IF ERL=390 THEN XL=-27*(
XL<0):GOSUB 870:S=S1:RESUME
550
940 IF YL>23 THEN RESUME 660
    ´ 1ｷﾞｮｳ ｱｷﾏﾏ ´
950 IF XL>27 THEN XL=0:GOSUB
 870:RESUME
```

●ジャンプのプログラム

●いろいろなサブルーチン

```
960 IF XL<0 THEN XL=27:GOSUB
 870:RESUME
970 IF XB>27 THEN XB=0:MOVE7
:RESUME
980 IF XB<0 THEN XB=27:MOVE7
:RESUME
990 IF XL+1>27 THEN XL=0:GOS
UB 870:RESUME
1000 IF XL-1<0 THEN XL=27:GO
SUB 870:RESUME
1010 IF X<12 THEN X=228:RESU
ME
1020 IF X>228 THEN X=12:RESU
ME
1030 END
```

●エラーが出たときの処理

マリオワールド変数リスト

X, Y	マリオのスプライト座標	R	リンゴを拾った数
XY, YL	マリオの足元のBG座標	Z	次に拾う数字のコード
XN, YN	ニタニタのスプライト座標	K$	ジャンプ台のBGキャラクタ
XE, YE	ニタニタの足元のBG座標	R$H	リンゴのBGキャラクタ
NN	ニタニタの数－1	H$	ハシゴのBGキャラクター
N I	マリオが衝突したニタニタの番号	U$	床のBGキャラクタ
D( )	ニタニタが動いた方向	S	マリオの方向、スプライト番号
Q( )	ニタニタが次に動く方向	V	マリオが動く方向
I	ニタニタのMOVE番号	K	マリオの速度、ジャンプの飛距離
W	得点	G	ジャンプの下向きの速度
T	制限時間	F	ジャンプ中のフラグ

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0																												
1																												
2			2														4					H72		H72		H72		
3	F00	F00	F00	H20						9						F00	F00	F00	K51	K51	F00	F00	F00	F00	F00	F00	F00	F00
4				H20			F00	F00	F00	F00	F00	F00	H20															
5		J30		H20									H20											J30			8	
6	F00	F00	F00	H20						J30			H20				H72		F00	F00	F00	H20	F00	F00	F00	F00	F00	F00
7				H20					F00	F00	F00	F00	H20			F00	F00	F00	F00			H20						
8				H20		H73	1						H20									H20				J30		
9	F00	H20	F00	F00	F00	F00	F00	H20					H20				J30					H20			F00	F00	F00	F00
10		H20						H20					H20	F00	F00	F00	F00	F00	F00			H20						
11		H20						H20					H20									H20						
12	5	H20				J30		H20					H20									H20					H73	
13	F00	F00	K51	K51	F00	F00	F00	F00	F00	H20			H20				J30			7		H20				H20	F00	F00
14										H20			H20			F00	F00	F00	F00	F00	F00	H20				H20		
15										H20			H20													H20		
16		H73								H20			H20		3	H73				H72				H73		H20	J30	J30
17	F00	F00	F00	H20						H20			H20	F00	F00	F00	K51	K51	F00	F00	K51	K51	F00	F00	F00	H20	F00	F00
18				H20						H20			H20													H20		
19		6		H20				0		H20						J30				J30				J30		H20		
20	F00	F00	F00	F00	F00	F00	F00	F00	F00	F00	F00	F00	F00	F00	F00	F00	F00	F00	F00	F00	F00	F00	F00	F00	F00	F00	F00	F00

4. スターキラー

A. 遊び方

プログラムを呼び出すときは，F4キーを押すか，または

GAME 3 RETURN

とキーを押してください。

プログラムが始まると，写真12のような画面になります。このゲームは，1人用と2人用が選べるようになっていますから，1人で遊ぶときは，キーボードのNキーを押し，2人で遊ぶときは，Yキーを押してください。1人で遊ぶときは，相手の宇宙船をコンピュータが操縦します。

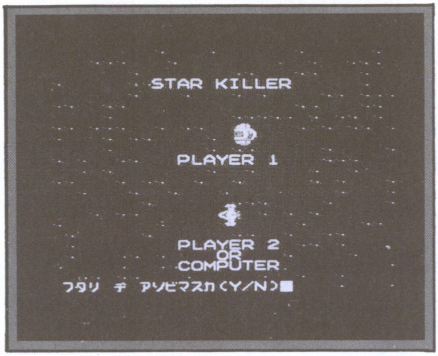
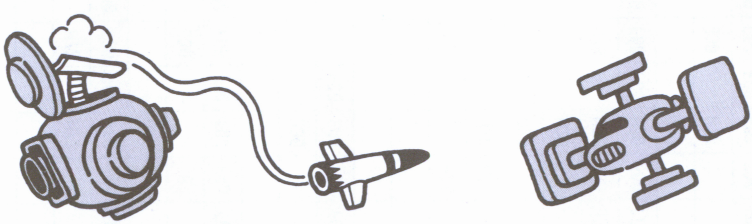


写真12

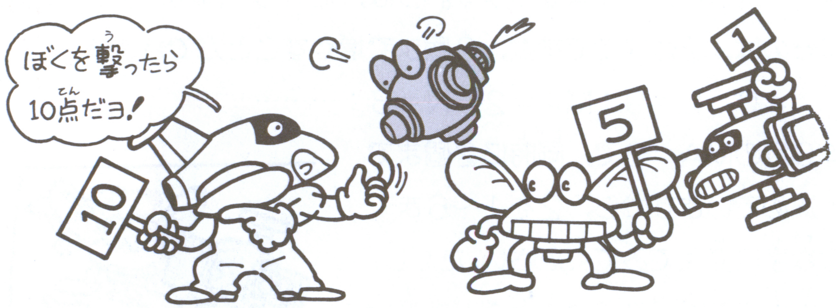
1人用ゲームか2人用ゲームかを決めると，ゲームが始まります。プレーヤーの動かす宇宙船は，ゲームが始まっても動き出しません。コントローラの✚ボタンを押してください。これで宇宙船が，それぞれの陣地から動き出しますので，コントロールしてください。



Aボタンを押すと，ミサイルが発射されます。このミサイルで相手の宇宙船や，そのほかの飛行体を撃ってください。相手の宇宙船に当たると10点，ファイターフライに当たると5点，そのほかの飛行体に当たると1点です。

Bボタンは，ブレーキです。Bボタンを押すと，宇宙船のスピードが遅くなりますが，そのかわり，こまわりがきくようになります。宇宙船のスピードを元に戻したいときは，もう1度Bボタンを押してください。

自分の宇宙船の数は，初めは4隻です。スピナーやファイアーボールなどの飛行体が，3個ずつ次々と現れ，ひと巡りすると，またスピナーに戻りますが，そのときに自分の宇宙船が，1隻増えます。ですから，得点の高い相手の宇宙船ばかりではなく，そのほかのキャラクタも，どんどんと撃ってください。



1人用ゲームのときは，自分の宇宙船がなくなったときに，ゲームオーバーになります。2人用ゲームのときは，2人とも宇宙船がなくなったときに，ゲームオーバーとなります。2人用ゲームで，1方の宇宙船がなくなったときには，もう1方の宇宙船がなくなってしまうまで，コンピュータが相手をします。

B．改造の方法

①　最初に持っている宇宙船の数を変えてみましょう

1人用ゲームでも2人用ゲームでも，プレーヤーの持っている宇宙船の数は，ゲームが始まったときには4隻ずつです。この数を変えてみましょう。2人用ゲームのときは，それぞれの宇宙船の数を変えて，ハンディをつけることができます。

```
40　L=9:MA=2:N=3:V=7:C=7:S(Ø)
=4:S(1)=4
```

プレーヤー1の宇宙船の数　　プレーヤー2の宇宙船の数

②　出てくる飛行体の数を変えてみましょう

宇宙船をコントロールしていると，スピナーやファイターフライなどの飛行体が，邪魔をします。今は，それらが同時に画面に出てくるのは，2個までですが，6個まで増やすことができます。

```
40　L=9:MA=2:N=3:………以下略
```

4Ø行のMA＝2で，同時に2個までと決めていますから，2を1～6までの数字に変えてみてください。ただし，3以上の数字にすると，ときどき飛行体が見えなくなることがあります。7以上の数にするとエラーになり，また，Øにしても1個は必ず出るようになっています。



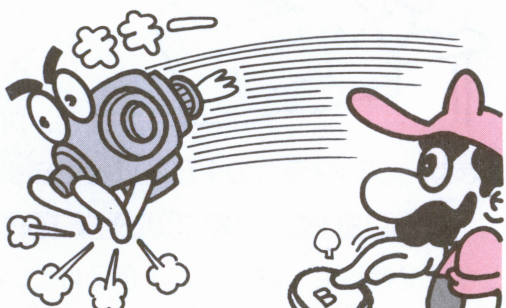
飛行体は，スピナーで始まって，3個命中させると，次の飛行体に変わるようになっています。
この数を変えてみましょう。

4Ø　L＝9：MA＝2：N＝3：V＝7：………以下略

今はN＝3になっています。この数字を1以上の適当な数に変えてみましょう。大きな数にすると，すべての飛行体が現れた後，元のスピナーに戻るまでの時間が，長くなくなります。そうすると，宇宙船が，1隻増えるのも遅くなります。これは，ゲームが難しくなったということです。

③　宇宙船のスピードを変えてみましょう

今のBボタンは，ブレーキボタンですから，押すと宇宙船のスピードが遅くなって，もう1度押すと，元に戻るようになっています。
これを改造して，Bボタンを押すと加速し，もう1度押すと元のスピードに戻るようにしてみましょう。速いスピードにしたいときに，試してみてください。



4Ø　L＝9：MA＝2：N＝3：V＝7：C＝7……以下略

このV＝7の数字を3にすると，Bボタンが加速ボタンに変わります。ただし，この改造をすると，Bボタンを押して加速したときに，宇宙船の速さが，ミサイルと同じになってしまいます。加速中にミサイルを発射すると，ミサイルを抱えて飛んでいるようになり，宇

宙船の方向を変えると，ミサイルだけが機体から切り離されて，元の進行方向に飛んでいくことになります。命中の判定などは，前と同じですから，攻撃の仕方を変えて楽しんでください。

数字を3ではなく，たとえば1Øくらいにすると，Bボタンは急ブレーキボタンになります。

④ 1人用・2人用を選ぶ画面をなくしてみましょう



このプログラムは，RUNすると必ず，1人用にするか2人用にするかの入力があるまで止まっています。1人で何回もプレイするときには，RUNするとすぐにゲームが始められるように，改造しましょう。これは大手術ですが，次のようにしてください。

まず，プログラムの1Ø7Ø行から後を，全部消してしまいます。

DELETE 1Ø7Ø- RETURN

とキーを押してください。1Ø7Øという数字を間違えないように注意してください。これで1Ø7Ø行から最後の行までが，まとめて消えました。

では，9Ø行を次のように変えてください。

```
9Ø GOSUB 1Ø7Ø
9Ø IF STRIG(Ø)=Ø THEN 9Ø
```

こうしておくと，RUNした後，Ⅰコントローラの A，B，START，

SELECTのどのボタンを押しても，すぐにゲームが始まるようになります。ゲームオーバーになったときも，どれかを押せばすぐにゲームが始まります。



最後に，新しく次の1行を追加してください。この行は，1人用にするか，2人用にするかで少し異なります。

（1人用） 85 M=1　　（2人用） 85 M=−1

この数字を1にしておけば，1人用になります。

【注意】 ここから後の,「邪魔ものをおいてみましょう」と「基地を作ってみましょう」を行なう人は，必ずプログラムの一部を，次のように改造してください。そうしないと，邪魔ものや基地が壊されてしまいます。

まず1ØØ行を次のように変えます。

```
1ØØ VIEW:FORP=ØTO2ØØ:……以下略
1ØØ VIEW
```

（全部消してしまう）

次に，1Ø1行～1Ø4行までを，新しく追加してください。

```
1Ø1 FOR P=Ø TO 1ØØ
1Ø2 X=RND(28):Y=RND(23):IF
SCR$(X, Y)<>" " THEN 1Ø2
1Ø3 LOCATE X, Y:PRINT CHR$(2
Ø5);
1Ø4 NEXT:P=Ø
```

（" " の間は1文字あける）

⑤ 邪魔ものをおいてみましょう

このプログラムでは，宇宙船の飛び回る宇宙空間に，邪魔ものをおくことができます。宇宙船が邪魔ものにぶつかると，爆発してしまいますから，ゲームがたいへん難しくなります。邪魔ものをおくときには，2つのルールがあります。次の説明をよく読んでから，試してください。

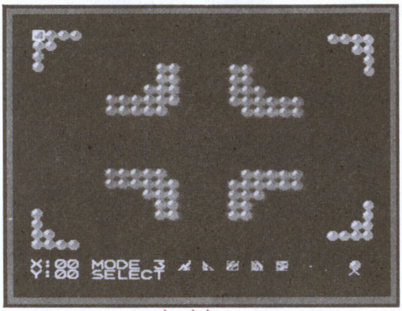


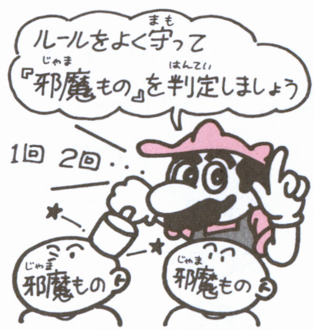
写真13

まず，STOP キーを押して，プログラムを止めて，

B G T O O L RETURN

とキーを押して，BGグラフィックモードにしてから，CLR HOME キーを6回，DEL キーを7回押してください。これで，写真13のような画面になります。このとき,「へ」マークがある「●」が，邪魔ものに使えるBGキャラクタです。これを使わないと，邪魔ものと判定してもらえません。これがルールその1です。

次に，このBGキャラクタをおくときの配色番号は，このままの「Ø」か，RETURN キーを3回押したときの「3」にしなくてはいけません。「Ø」か「3」以外の配色番号にすると，やはり邪魔ものと判定してもらえません。これがルールその2です。



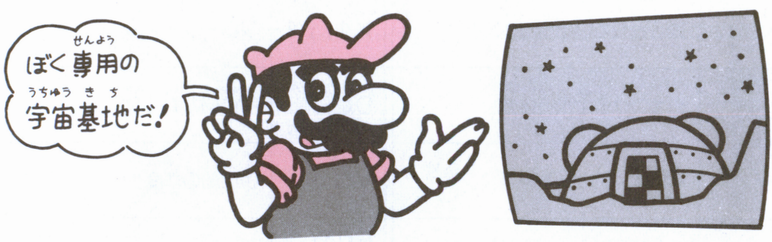
この2つのルールを守れば，画面上のどこにおくかは自由です。(ただし，宇宙船のスタート地点に邪魔ものをおくと，ゲームを始め

ることができません。うまくデザインすると，ゲームがとてもおもしろくなりますから，いろいろと試してください。

邪魔もののおき方によっては，宇宙船の操縦が，とても難しくなります。でも，邪魔ものは，ミサイルで消すことができます。得点にはなりませんが，ゲームが始まったら，邪魔ものをどんどん撃っていくのも，ひとつの作戦になります。ただし，宇宙船以外のアニメキャラクタは，邪魔ものにぶつかっても爆発しません。

⑥　基地を作ってみましょう

それぞれ自分専用の基地を作りましょう。基地を作るときも，BG TOOL命令で，BGグラフィックにします。この場合も，2つのルールがあります。BGグラフィックモードにする方法と，ルールその1は，「邪魔ものをおいてみましょう」と同じです。



ルールその2は，プレーヤー1とプレーヤー2の基地を作るときの配色番号です。

プレーヤー1の基地………配色番号　1　で作る
プレーヤー2の基地………配色番号　2　で作る

と決まっていますから，配色番号を間違えないようにしてください。基地のかたちや，どこに作るかは自由にできます。

それぞれの基地を作ると，

配色番号１の基地………プレーヤー２がぶつかると爆発
配色番号２の基地………プレーヤー１がぶつかると爆発

のようになりますから，危なくなったら自分の基地に逃げ込みます。このとき，Ｂボタンの急ブレーキを使って作戦に変化をつけてください。

ミサイルの攻撃を受けると基地はこわれてしまいます。基地と邪魔ものとを組み合わせておけば，さらに楽しさが広がります。

スターキラー変数リスト

T	STRIG（ ）の値	V	Ｂボタンを押したときの速度
S	STICK（ ）の値	X0，Y0	プレーヤー１のホームポジション
U	表示されている飛行体の数	X1，Y1	プレーヤー２のホームポジション
N	やっつける全飛行体数	D（ ）	宇宙船が動く方向
C	飛行体のアニメキャラクタ番号	V（ ）	宇宙船の速度
MA	一画面に表示する飛行体の数	F（0）〜F（7）	何と何がぶつかったかのフラグ
L	コンピュータの強さ	F（8）	ファイターフライの出現フラグ
M	１人ゲームか２人ゲームか	SC（ ）	スコア
P	0：Ⅰコントローラ	O	攻撃したMOVE番号など
	1：Ⅱコントローラ	D	攻撃されたMOVE番号など

スターキラープログラムリスト

```
10 '"AREA 88 V7.3"
20 T=O=U=C=H=D=I=K=L=M=N=P=S
=V=X=Y=MA=X0=X1=Y0=Y1=PP=Q
30 DIM D(1),V(1),F(8),SC(1),
S(1)
40 L=9:MA=2:N=3:V=7:C=7:S(0)
=4:S(1)=4
50 X0=70:Y0=80:X1=110:Y1=60
60 D(0)=3:V(0)=2:D(1)=7:V(1)
=2:F(0)=3:F(1)=3:MA=MA+1:U=N
:M=1
70 SPRITE ON:CGSET 1,1:PLAY"
T1Y2M1V9O3R1":PALETS 1,13,48
,38,18
80 FOR I=15 TO 18:A$=A$+SCR$
(I,22):NEXT:H=VAL(A$)
```

●準備

```
90 GOSUB 1070
```

●１人用か２人用を選ぶサブルーチンを呼ぶ

```
100 VIEW:FORP=0TO200:LOCATER
ND(27),RND(23):PRINTCHR$(205
);:NEXT:P=0
```

●ＢＧを呼び出して星を描く

```
110 '
120 I=P+6:IF F(P) THEN ON F(
P) GOSUB 670,700,680:GOTO 15
0
130 IF M=P THEN GOSUB 390:GO
TO 150
140 GOSUB 210
150 IF MOVE(I)=0 THEN ERA I
160 GOSUB 530
170 GOSUB 800
180 ON F(8)+1 GOSUB 940,1040

190 P=1-P:GOTO 120
200 '
```

●メインルーチン

```
210 S=STICK(P):T=STRIG(P)
220 IF T<>4 THEN F(I)=1
230 IF S+T=0 THEN RETURN
240 IFS=0THEN350
250 S=-1*(S=8)-3*(S=1)-5*(S=
4)-7*(S=2)-2*(S=9)-4*(S=5)-6
*(S=6)-8*(S=10)
260 Q=D(P)
270 IF S=Q THEN 350
280 IF Q>4 THEN 310
```

※本書リストに出てくる「∗」は，「＊」と同じ文字です。画面に出てくる文字と，多少異なりますので，ご了承ください。

```
290 IF (S>Q AND S<=Q+4) THEN
 320
300 GOTO 330
310 IF (S>=Q-4 AND S<Q)THEN
330
320 Q=Q+1:GOTO 340
330 Q=Q-1
340 PP=-1*(Q>8)-8*(Q<1):Q=-Q
*(PP=Q)+PP:D(P)=Q
350 IF T AND4 THEN IF F(I) T
HEN V(P)=V-V(P):F(I)=0
360 DEF MOVE(P)=SPRITE (P+8,
D(P),V(P),255,1):MOVE P
370 IF T>7 THEN DEF MOVE(I)=
SPRITE(12,D(P),1,80,0,P):POS
ITION I,(XPOS(P)+4)MOD256,(Y
POS(P)+4)MOD256:MOVE I
380 RETURN
390
400 K=K-1:IF K>0 THEN RETURN

410 T=0:D=0:X=(XPOS(1-P)-XPO
S(P))/13:Y=(YPOS(1-P)-YPOS(P
))/13
420 IF X>0 THEN D=3
430 IF X<0 THEN D=7
440 IF Y>0 THEN D=D+5
450 IF Y<0 THEN D=D+1:IF X<0
 THEN D=16
460 IF X AND Y THEN D=D/2
470 S=2:O=D-D(P):IF(0<O AND
O<5)OR O<-3 THEN S=1
480 IF O=0 THEN S=0
490 IF V(1-P)<>V(P) THEN T=4

500 IF MOVE(I)=0 OR RND(10)=
0 THEN T=T+8
510 IF X*X+Y*Y>200 THEN RETU
RN
520 K=RND(L):GOTO 220
530
540 D=CRASH(0):IF D>0 AND D<
6 THEN O=0:GOSUB 590
550 D=CRASH(1):IF D>1 AND D<
6 THEN O=1:GOSUB 590
560 D=CRASH(6):IF D>0 AND D<
6 THEN O=6:GOSUB 610
570 D=CRASH(7):IF D=0 OR (D>
```

●✛ボタンやトリガボタンが押(お)された判定(はんてい)

●1人用(ひとりよう)のとき相手(あいて)の宇宙船(うちゅうせん)を動(うご)かす

```
1 AND D<6)THEN O=7:GOSUB 610

580 RETURN
590 X=ABS(XPOS(O)-XPOS(D)):Y
=ABS(YPOS(O)-YPOS(D)):IF X<1
0 AND Y<10 THEN IF MOVE(O) A
ND MOVE(D) AND F(D)=0 AND F(
O)=0 THEN F(O)=1:F(D)=1:IF D
>1 THEN 650
600 RETURN
610 X=ABS(XPOS(O)-XPOS(D)-4)
:Y=ABS(YPOS(O)-YPOS(D)-4):IF
 X<8 AND Y<8 THEN IF MOVE(O)
 AND MOVE(D) AND F(D)=0 THEN
 ERA O:F(D)=1:GOTO 630
620 RETURN
630 IF D<2 THEN SC(O-6)=SC(O
-6)+10:RETURN
640 SC(O-6)=SC(O-6)+1+F(8)*4

650 DEF MOVE(D)=SPRITE(10,RN
D(9),RND(3)+4,10):MOVE D:PLA
Y"O4C":RETURN
660
670 F(P)=2:DEF MOVE(P)=SPRIT
E(10,D(P),V(P),70/V(P),0):MO
VE P:PLAY"O"+HEX$(P+3)+"GG":
S(P)=S(P)-1:RETURN
680 IF P=M OR STRIG(P)+STICK
(P) THEN F(P)=0:POSITION P,X
0+P*X1,Y0+P*Y1
690 RETURN
700 IF MOVE(P) THEN RETURN
710 F(P)=3:ERA P
720 IF SC(0)>H AND M<>0 THEN
 H=SC(0)
730 IF SC(1)>H AND M<>1 THEN
 H=SC(1)
740 IF L<1 THEN L=5
750 LOCATE 0,20:PRINT"SCORE
",SC(0),SC(1),"LEVEL":PRINT"
SHIPS",S(0),S(1),L;" ":PRINT
"HI-SCORE",H,
760 IF S(P)>0 THEN RETURN
770 IF M=P THEN L=L-1:S(P)=4
:GOTO 740
780 IF M=-1 THEN M=P:PRINT P
;" COM":PLAY"EE":RETURN
```

●どのアニメキャラクタ同志がぶつかったかなどの細かい判定をする

●宇宙船を爆発させる

●宇宙船が消えた後，コントローラのボタンが押されたら，次の宇宙船を出す

●ゲームオーバの表示をして再ゲーム

```
790 LOCATE 10,10:PRINT "GAME
 OVER":PLAY"O1C5CDED6C4C5":P
AUSE 200:RUN
```

```
800 '
810 X=XPOS(P)/8-1:Y=YPOS(P)/
8-2
820 IF X<0 OR X>27 OR Y<0 OR
 Y>23 THEN 880
830 IF SCR$(X,Y)<>"●"THEN 88
0
840 S=ASC(SCR$(X,Y,1))
850 IF S=P+1 THEN 880
860 IF F(P)=0 THEN F(P)=1
```

●宇宙船が敵の基地などにぶつかった判定

```
870 '
880 IF MOVE(I)=0 THEN RETURN

890 X=(XPOS(I)-12)/8:Y=(YPOS
(I)-20)/8
900 IF X<0 OR X>27 OR Y<0 OR
 Y>23 THEN RETURN
910 IF SCR$(X,Y)<>"●" THEN R
ETURN
920 S=ASC(SCR$(X,Y,1)):IF S=
P+1 THEN RETURN
930 ERA I:LOCATE X,Y:PRINT "
 ";:RETURN
```

●レーザーが敵の基地などに当たった判定

```
940 '
950 FOR I=2 TO MA:IF MOVE(I)
OR F(I)=2 THEN 980
960 IF F(I)=1 THEN GOSUB 990
:GOTO 980
970 DEF MOVE(I)=SPRITE (C,RN
D(9),RND(3)+C/2,80,0):MOVE I

980 NEXT:RETURN
```

●スピナーなどを出して動かす

```
990 '
1000 F(I)=0:ERA I:POSITION I
,120,120:IF U<MA THEN F(I)=2

1010 U=U-1:IF U=0 THEN F(2)=
0:F(8)=1:RETURN
1020 IF U<0 THEN U=N:F(2)=0:
F(3)=0:F(4)=0:F(5)=0:F(8)=0:
C=C-2:L=L-1:IF C<2 THEN C=7:
IF M>-1 THEN S(1-M)=S(1-M)+1
 '
1030 RETURN
```

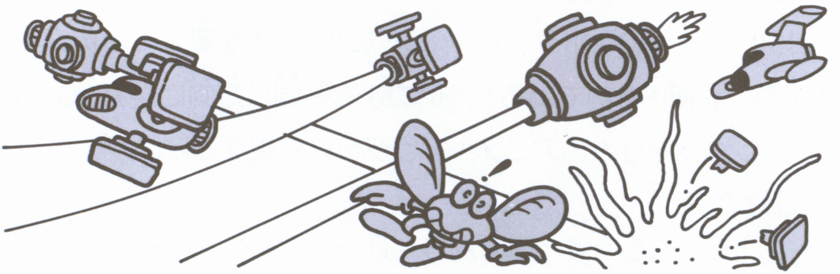
●スピナーなどを命中させた後の処理

```
1040 IF F(2) THEN I=2:GOTO99
```

```
0
1050 IF MOVE(2)=0 THEN DEF M
OVE(2)=SPRITE(2,RND(8)+1,RND
(3)+1,10,0):MOVE 2
1060 RETURN
1070 '
1080 VIEW:RESTORE 1070
1090 FORP=0TO200:LOCATERND(2
7),RND(23):PRINTCHR$(205):NE
XT
1100 LOCATE8,3:PRINT"STAR KI
LLER"
1110 FOR P=0 TO 1:LOCATE 10,
10+P*8:DEF MOVE(P)=SPRITE(P+
8,D(P),2,30):POSITION P,X0+P
*X1,Y0+P*Y1:MOVE P
1120 IF MOVE(P) THEN 1120
1130 READ A$:FOR I=1 TO LEN(
A$):PRINT MID$(A$,I,1);:PLAY
"C":NEXT:NEXT
1140 FOR P=2 TO 3:LOCATE 10,
17+P:READ A$:FOR I=1 TO LEN(
A$):PRINT MID$(A$,I,1);:PLAY
"C":NEXT:NEXT:LOCATE1,22:PRI
NT"ﾌﾀﾘ ﾃﾞ ｱｿﾋﾞﾏｽｶ(Y/N)";:M=-
1
1150 A$=INKEY$(0):IFA$="N"TH
ENM=1
1160 PAUSE80:RETURN
1170 DATA "PLAYER 1","PLAYER
 2","   OR","COMPUTER"
```

●ファイターフライを出して動かす

●１人用２人用を選ぶサブルーチン



DEF MOVEとDEF SPRITE

ファミリーベーシックでは，アニメキャラクタを画面で動かすために，2とおりの方法があります。1つはファミリーベーシックの大きな特徴であるDEF　MOVE命令を使う方法です。この方法だと，短いプログラムでアニメキャラクタに自然な動きをさせることができます。もう1つはDEF　SPRITE命令を使う方法で，これは好みのキャラクタに自由な動きをさせることができます。

ここでは，その2つの方法でアニメキャラクタを動かすプログラムを紹介しましょう。2つのプログラムを見比べると，DEF　MOVEとDEF　SPRITEのそれぞれの特徴がわかっていただけると思います。



1.DEF　MOVE（リスト2）

プログラムをRUNさせると，ペンペンが出てきて歩き出します。Iコントローラの✚ボタンで，ペンペンは向きを変えて進みます。✚ボタンを押さないでいると，いつまでも同じ方向に進み続けます。ペンペンをほかのアニメキャラクタに変更するには，

11Ø行の

　　11Ø　AC=4　：　AS=Ø……以下略

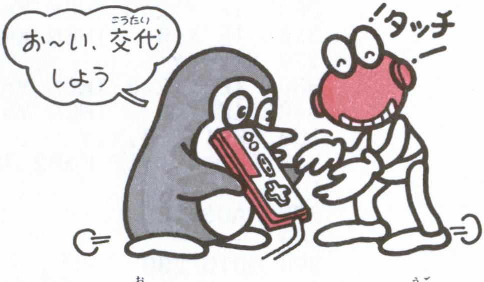
数字をØ～15にすると，それに応じたアニメキャラクタになります。

```
10 'LIST 2
100 CLEAR : CLS : SPRITE ON
: CGSET 1,1
110 AC=4 : AS=0 : D=5 : X=11
6 : Y=116
120 POSITION 0,X,Y
200 '==========
210 S=STICK(0) : IF S=0 GOTO
 360
220  IF S=8 THEN D=1
230  IF S=9 THEN D=2
240  IF S=1 THEN D=3
250  IF S=5 THEN D=4
260  IF S=4 THEN D=5
270  IF S=6 THEN D=6
280  IF S=2 THEN D=7
290  IF S=10THEN D=8
300 IF AS=D GOTO 360
310 AS=D
320 X=XPOS(0) : Y=YPOS(0)
330 DEFMOVE(0)=SPRITE(AC,D,1
,255,0,0)
340 POSITION 0,X,Y
350 MOVE 0
360 '==========
370 IF MOVE(0)<>0 GOTO 410
380 DEFMOVE(0)=SPRITE(AC,D,1
,255,0,0)
390 POSITION 0,X,Y
400 MOVE 0
410 GOTO 200
```

リスト２

2.DEF（デファイン） SPRITE（スプライト）（リスト３）

ペンペン以外（いがい）のアニメキャラクタにしたいときは，図（ず）９のように，１２０行（ぎょう）の数字（すうじ）を変（か）えてください。



※RUN（ラン）した後（あと），Ｉコントローラの✚ボタンを押（お）すと，アニメキャラクタが動（うご）きます。

```
120 P=96:P1=P:P2=100
```

96		100
0	マリオ	8
28	レディ	36
56	ファイターフライ	60
64	アキレス	68
88	ニタニタ	92
112	ファイアーボール	116
144	スピナー	148
184	カメさん	188
192	カニさん	196
200	トリさん	204

図9

```
10 'LIST 3
100 CLEAR : CLS : CGSET 1,1
: SPRITE ON
110 X=116 : Y=116 : F=1
120 P=96 : P1=P : P2=100
200 '==========
210 K=STICK(0) : IF K=0 GOTO
 380
220  AX=0 : AY=0
230  IF K=1 OR K=9 OR K=5 TH
EN AX=4 : F=1
240  IF K=2 OR K=6 OR K=10 T
HEN AX=-4 : F=-1
250  IF K=4 OR K=5 OR K=6  T
HEN AY=4
260  IF K=8 OR K=9 OR K=10 T
HEN AY=-4
270 IF F=1 THEN DEF SPRITE 0
,(0,1,0,1,0)=CHR$(P+1)+CHR$(
P)+CHR$(P+3)+CHR$(P+2)
280 IF F=-1 THEN DEF SPRITE
0,(0,1,0,0,0)=CHR$(P)+CHR$(P
+1)+CHR$(P+2)+CHR$(P+3)
290 X=X+AX
300  IF X>240 THEN X=0
310  IF X<0   THEN X=240
320 Y=Y+AY
330  IF Y>240 THEN Y=0
340  IF Y<0   THEN Y=240
350 SPRITE 0,X,Y
360 P=P+4 : IF P>P2 THEN P=P
1
370 PAUSE 3
380 '-----
390 GOTO 200
```

リスト3

4. FAMILY BASIC™ V3の命令早わかり

ファミリーベーシック

めいれいはや



本文の読み方

ここでは，ファミリーベーシックV3で新たに追加されたBASICの命令語について，解説します。解説文の読み方は，以下のようになります。

働　き……命令語の働きをひとことで説明したものです。

省略形……命令語を短縮したものです。

文　法……命令語の使い方を一般的な形で表したものです。

- 〔　〕の中は，ユーザーが自由に省略できます。
- ｛　｝の中の項目は，ユーザーが自由に選択できますが，必ずどれかを選ばなくてはなりません。
- （　），「,（カンマ）」，「:（コロン）」，「;（セミコロン）」，「-（ハイフン）」，「=（イコール）」などの記号は，文法どおりに入力します。

説　明……命令の使い方や注意点などが詳しく説明してあります。

……命令を使った例題や実行結果などが示されています。

……命令語の項目を省略したとき，自動的にセットされる値のことです。

〔注意〕TABとSPCは，拡張用予約語のため，変数名には使えません。

RETURN キーを押してその行を飛ばし（スキップ）ます。文字を入力してから RETURN キーを押すと，その行は新しい内容のものと入れ替わります。 STOP キーを押すと，AUTO 命令は終了します。

BACKUP

働き プログラムやBG面をバックアップするときに使います。

省略形 BA.

文法 BACKUP

説明 本体の電源を OFF にしても，ユーザー RAM 領域の内容が消えてしまわないように，バッテリー（電池）により内容を残しておくための命令です。

プログラムと一緒にBG面1をバックアップしたい場合は，BG GET 命令を使ってあらかじめ，ユーザー RAM 領域に転送しておく必要があります。

ただし，この場合はメモリ内のプロ

AUTO（オート）

働き　入力するプログラムの行番号を自動的に付けてくれます。

省略形　A.

文法　AUTO〔m〕,〔n〕

m…始めの行番号（Ø～65534）

n…いくつずつ行番号を増加していくか（Ø～65534）

デフォルト値　m＝1Ø　n＝1Ø

説明　これから入力するプログラムの始めの行番号と，いくつずつ増加させて行番号を付けるのか，その値を指定すると，自動的に行番号が付けられます。したがって，入力するのは，自動的についた行番号のあとのプログラム部分だけになります。１行入力が終わって RETURN キーを押すと，指定した増加分で次の行番号が付けられ表示されていきます。

m，nの指定を省略した場合は，デフォルト値として1Øが選択されます。

プログラム中にすでにある行番号が表示された場合，前のプログラムを残すときは，

```
ｻﾝﾌﾟﾙ ﾌﾟﾛｸﾞﾗﾑ
** AUTO **
AUTO
OK
10 REM *AUTO
20 X=25
30 PRINT X
40            ←ここでストップキーをおす
OK
AUTO 100
OK
100 REM *AUTO
110 X=25
120 PRINT X
130           ←ここでストップキーをおす
OK
AUTO 100,30
100 REM *AUTO
130 X=25
160 PRINT X
190           ←ここでストップキーをおす
OK
AUTO , 50
190 REM *AUTO
240 X=25
290 PRINT X
340           ←ここでストップキーをおす
OK
```

グラムが3Kバイト以下でなくてはなりません(BGGETの項目参照)。命令実行後，バックアップスイッチをONにするようメッセージが表示されます。

ビージーゲット
BGGET

働き BG面1をバックアップするための命令です。

省略形 BGG.

文法 BG GET

説明 V RAMにあるBG面1のデータが，ユーザーRAMに転送されます。ただし，ユーザーRAMにメモリの余裕がない場合は，OMエラーが発生します(メモリ容量は4Kバイト)。

ビージープット
BGPUT

働き BG GETによりバックアップされたBG面を，V RAMへ転送する命令です。

省略形 BGP.

文法 BG PUT

説明 BG GET によりユーザー RAMに転送されたデータを，V RAM に転送します。すでに転送済みの場合は，NB エラーが発生します。
また，HOT START の場合は，自動的に転送され，BG PUT 実行後と同じ状態になります。

BGTOOL

働き 背景画を描くための，モードに移行する命令です。

省略形 BG.

文法 BGTOOL

説明 BG面１で背景画を描くために，BASIC モードからBGグラフィックモードへ移行します。これは，プログラムを作成している途中でも実行することができます。また，この命令をプログラムに組み込むこともできます。BGグラフィックの作成方法は，これまでと同じです。なお，

BASICモードに戻るには，ESCキーを押した後，STOPキーを押します。

CAN

働き DEF MOVEで動きを定義したアニメキャラクタを，未定義にします。

文法 CAN n0，n1，n2，……n7
n0～n7…スプライト番号

説明 DEF MOVEによって定義したアニメキャラクタの，表示と座標位置の両方を同時に未定義にする命令です。
これまでのファミリーベーシックでは，アニメキャラクタの表示を消すとともに，座標位置を未定義(0，0)にするには，ERA命令とPOSITION命令の2つが必要でした。V3では，このCAN命令だけで，これを実行することができます。

CLICK

働き キー入力音（クリック音）のON/OFFを行ないます。

省略形 CLI.O.／CLI.OF.

文法 CLICK { ON …クリック音発生 / OFF …無音 }

- CLICK ON…クリック音発生
- CLICK OFF…無音

CLS

働き BG面の表示をクリアします。

省略形 CL.

文法 CLS〔BG面〕

Ø…BG面Øのクリア

1…BG面1のクリア

ただし，省略の場合は現在アクセス中のBG面がクリアされる。

説明 BG面には，Øと1の2面があります。どちらの画面をクリアするかを番号で指定します。

CRASH（クラッシュ）

働き DEF MOVEによるアニメキャラクタ同志の重なりを判定します。

省略形 CR.

文法 CRASH（n）

n…スプライト番号(0～7)

説明 アニメキャラクタ同志の重なりを判定します。

アニメキャラクタが重なっている場合，CRASH(n)は，重なっているスプライトのうちで，いちばん小さなスプライト番号となります。

ただし，重なっていない場合は，－1となり，またスプライトが未定義の場合は，－2となります。

```
1  ' ｻﾝﾌﾟﾙ ﾌﾟﾛｸﾞﾗﾑ
5  ' SPRITE
10 VIEW
20 SPRITE ON
30 CGSET 1,0
40 D=3
50 FOR N=0 TO 7 :DEF MOVE(N)
=SPRITE(N,D,N+1,255,0):NEXT
60 MOVE 0,1,2,3,4,5,6,7
70 LOCATE 0,3 :PRINT"VCT(0)"
;VCT(0)
80 LOCATE 0,5 :PRINT"XPOS(0)
,YPOS(0)" ; XPOS(0) ; "," ;Y
POS(0) ; "  "
90 LOCATE 0,7 :PRINT "CRASH(
0) :" ;CRASH(0)
100 LOCATE 0,1 :PRINT "MOVE(
0) :" ;MOVE(0)
110 IF MOVE(0)=-1 THEN 70
120 D=RND(8)+1 :PLAY"T1O4B1A
G2FE"
130 GOTO 50
```

DELETE（デリート）

働き プログラムの中で消したい部分があるとき，その行番号を指定することと

によって，まとめて取り消すことができます。

省略形 DEL.

文法 DELETE〔m〕{'-'}〔n〕

m…取り消しを始める行番号

n…取り消しを終える行番号

説明 mの値だけを指定した場合は，その行だけが取り消されます(②)。また，－(ハイフン)からnの値だけを指定した場合は，プログラムの先頭から指定行(n)までが，取り消されます(③)。nの値を省略した場合は，mの値から最後の行までが，取り消されます(④)。

DELETE　m－n　…①

DELETE　m　…②

DELETE　－n　…③

DELETE　m－　…④

```
ｻﾝﾌﾟﾙ ﾌﾟﾛｸﾞﾗﾑ
** DELETE **
LIST
10 REM *DELETE
20 INPUT "A,B,X" ;A,B,X
30 INPUT "A,B,X" ;A,B,X
40 Y=A*X+B
50 Y=A*X+B
60 Y=A*X+B
70 PRINT "A*X+B=" ;Y
80 PRINT "A*X+B=" ;Y
90 PRINT "A*X+B=" ;Y
100 END
OK
DELETE 30,50
OK
LIST
10 REM *DELETE
20 INPUT "A,B,X" ;A,B,X
60 Y=A*X+B
70 PRINT "A*X+B=" ;Y
80 PRINT "A*X+B=" ;Y
90 PRINT "A*X+B=" ;Y
100 END
OK
DELETE 80,
OK
LIST
10 REM *DELETE
20 INPUT "A,B,X" ;A,B,X
60 Y=A*X+B
70 PRINT "A*X+B=" ;Y
OK
```

イーアールエル
ERL

働き エラーの発生した行番号が求められます。

説明 エラーが発生したとき，どの行で発生したのか，その行番号を知るために使われます。
エラーが発生すると，ON ERROR GOTO 文によってエラー処理ルーチンに移り，ERL（エラーの起きた行番号）の値によって処理を決定するというように用いることができます。

ERR

働き エラーが発生したとき，そのエラーコードが求められます。

文法 ERR

説明 エラーが発生すると，ON ERROR GOTO 文によってエラー処理ルーチンに移り，その中でエラーの処理が行なわれます。
ERRは，エラーの内容に対応したエラーコードがその値となり，エラーの種類により，処理内容を決定す

るために使(つか)われます。
※ON(オン) ERROR(エラー) GOTO(ゴーツー)の項(こう)のサンプルプログラムおよびエラーコード表(ひょう)(P.127)参照(さんしょう)

ERROR(エラー)

働(はたら)き 仮(かり)のエラーを発生(はっせい)させます。

省略形(しょうりゃくけい) ERR.

文法(ぶんぽう) ERROR(エラー)　エラーコード

※エラーコード表(ひょう)(P.127)参照(さんしょう)

説明(せつめい) これは，エラー処理(しょり)ルーチンが正(ただ)しく働(はたら)くかどうかを確認(かくにん)するために，仮(かり)のエラーを発生(はっせい)させる命令(めいれい)です。指定(してい)したエラーコードに対応(たいおう)するエラーが，その行番号(ぎょうばんごう)で発生(はっせい)します。

FIND(ファインド)

働(はたら)き プログラムの中(なか)から，指定(してい)した文字(もじ)列(れつ)を探(さが)し，それを含(ふく)む行(ぎょう)を表示(ひょうじ)させる命令(めいれい)です。

省略形 FI.

文法 FIND "X"

X…探したい単語　31字以内

説明 これは，1度入力した行に対して行なうことのできる命令です。プログラムを変更するときなどにこの命令を使うと，変更したい単語（文字列といいます）を含む行をすべて表示してくれるので，とても便利です。

サンプル プログラム

```
** FIND **
LIST
10 REM *SEARCH
20 INPUT A,B,X
30 Y=A*X+B
40 PRINT A;B;X;Y
50 END
OK
FIND "A"
10 REM *SEARCH
20 INPUT A,B,X
30 Y=A*X+B
40 PRINT A;B;X;Y
OK
FIND " A"
20 INPUT A,B,X
40 PRINT A;B;X;Y
OK
```

FILTER

働き BG面の全面着色が実行されます。

省略形 FIL.

文法 FILTER　カラー番号

Ø番	1番	2番	3番	4番	5番	6番	7番
無色	赤	緑	黄	青	マゼンタ	空色	白

ただし，デフォルト値はØ

説明 フィルターをかけたように，淡く着色されます。

色の指定番号は，表のとおり0～7の8種類です。何も番号を指定しない場合は，デフォルト値（自動的にセットされる値）として，0が選択されます。

また，CTRとDキーを同時に押してもデフォルト状態になります。

GAME

働き ROMに内蔵されているゲームが実行されます。

省略形 GA.

文法 GAME P〔,Sw〕

P…ゲーム番号 0～3

Sw…背景画の転送を行なう：1

転送しない：0または省略

説明 ゲームは4種類あります。呼び出すときは，それぞれのゲーム番号で指定します。

この命令は，プログラム中に組み込むことはできません。

なお，GAMEを呼び出すと，その前に記憶されていたプログラムは，自動的に消えてしまいますから，注意してください。

V3カセットに内蔵されているゲームのうち，背景画の転送の必要があるのは，『マリオワールド』だけです。

'INSTR

働き プログラム中の文字列1の中で，指定した文字(文字列2)の位置が求められます。

省略形 INS.

文法 INSTR([P,]文字列1，文字列2)

P…文字列1を探し始める位置。
デフォルト値は1。

説明 まず，文字列1の中に文字列2が含まれているかどうかを探します。含まれていればその位置を，文字列1の先頭から数え，その数字が値とな

サンプル プログラム

```
5  REM *INSTR
10 A$="NINTENDO HUBASIC"
20 PRINT INSTR(A$,"TEN")
RUN
4
```

ります。値の範囲は1～31で，文字列がみつからなかった場合の値はØとなります。また，位置の数え方は左から右です。

LOADS

働き カセットテープからBGグラフィックのデータを呼び出す命令です。

文法 LOADS "ファイル名"

ただし，ファイル名：31文字以内

説明 プログラムをロードする命令（LOAD）と，基本的には同じです。
ファイル名を指定すると，その名前のBGが見つかるまで，他のBGやプログラムを読み飛ばし（SKIP と表示され）ます。
ファイル名を省略すると，カセットテープの1番目に入っているBGが呼び出されます。
なお，プログラムとBGを一緒に呼び出したい場合は，
LOAD "プログラム名":LOA

D S " ファイル名 " RETURN

とします。

ロード中は，画面に LOADING ファイル名と表示され，呼び出しが終了すると OK となります。

ON ERROR GOTO

働き エラーが発生したとき，分岐する行を指定します。

省略形 O. ERR. G.

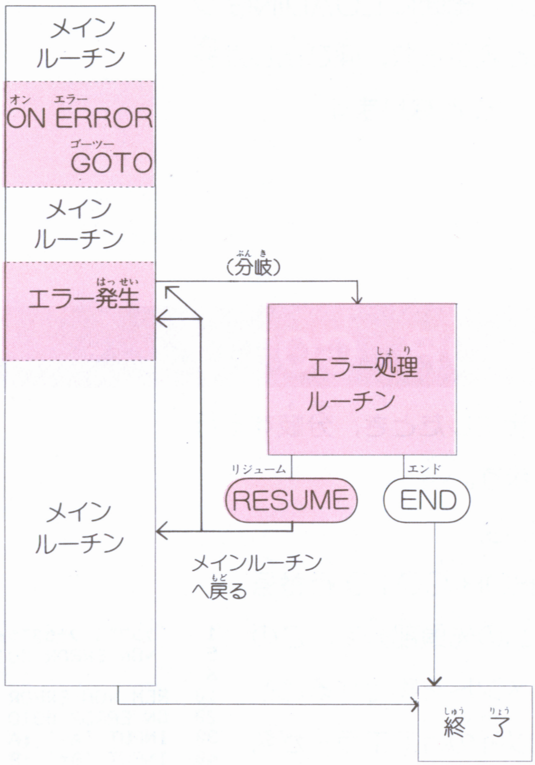
文法 ON ERROR GOTO 行番号

説明 プログラムの先頭部分で，このステートメントを実行することにより，以降の行でエラーが発生した場合，指定行（エラー処理ルーチン）へ分岐します。

分岐先ルーチンの終了行は，RESUME または END 命令にしておきます。RESUME の場合は，メインルーチンのあらかじめ指定しておいた行に再び戻ります。また，END の場合は，

```
1    'ｻﾝﾌﾟﾙ ﾌﾟﾛｸﾞﾗﾑ
5    '*ON ERROR GOTO/RESUME/
6    '                 ERR/ERL
10   REM *ON ERROR SAMPLE
20   ON ERROR GOTO 100
30   INPUT "A=" ;A
40   INPUT "B=" ;B
50   C=A/B
60   PRINT "A/B=" ;C
70   GOTO 30
100  PRINT "ﾄｹﾏｾﾝ"
110  PRINT "ERR=" ;ERR
120  PRINT "ERL=" ;ERL
130  RESUME 30
```

プログラムは終了し，コマンドの入力待ちとなります。



RENUM

働き プログラムの行番号を付け替える命令です。

省略形 REN.

文法 RENUM〔L〕,〔m〕,〔n〕

L…新しくつける開始行番号

m…変更を始める行の元の行番号

n…いくつごとに行番号を増加していくのか。

デフォルト値　L, m, n 共に10

説明 プログラムの行番号を整理する命令です。行番号が整理されるだけでなく，GOTO, GOSUB 文などの分岐先行番号も，新しい行番号に対応して自動的に変更されます。

ただし，ERL=1000のように，定数で指定した行番号は変更されませんので注意してください。

```
ｻﾝﾌﾟﾙ ﾌﾟﾛｸﾞﾗﾑ
** RENUM **
5  REM *RENUM
20 FOR I=0 TO 10
21 PRINT"BASIC V3"
55 NEXT
RENUM
OK
LIST
10 REM *RENUM
20 FOR I=0 TO 10
30 PRINT"BASIC V3"
40 NEXT
OK
RENUM 100
OK
LIST
100 REM *RENUM
110 FOR I=0 TO 10
120 PRINT"BASIC V3"
130 NEXT
OK
RENUM 20 , , 20
OK
LIST
20 REM *RENUM
40 FOR I=0 TO 10
60 PRINT"BASIC V3"
NEXT
OK
```

RESUME

働き エラー分岐ルーチンからメインルーチンに戻す命令です。

省略形 RESU.

文法 RESUME [{行番号 / NEXT}]

RESUME のみ …エラー発生行から再び実行

行番号 …指定した行番

号から実行

NEXT …エラー発生行の次の行から実行

説明 ON ERROR GOTO文によって分岐したエラー処理ルーチンから，メインルーチンに戻す命令です。

※ON ERROR GOTOの項のサンプルプログラム参照

SAVES

働き カセットテープにBGグラフィックのデータを保存する命令です。

文法 SAVES "ファイル名"

ただし，ファイル名 31文字以内

説明 プログラムをセーブする命令（SAVE）と基本的には同じです。

プログラムとBGを一緒に保存しておくには，

SAVE“プログラム名”：SAVES“ファイル名” RETURN

とします。

セーブ中は，画面にWRITING ファイル名と表示され，読み込みが終了するとOKとなります。

SCREEN

働き 表示面とアクティブ面に，2面あるBG面のいずれかを割り当てます。

省略形 SC.

文法 SCREEN 表示面〔，アクティブ面〕

命令	表示面	アクティブ面
SCREEN Ø，Ø	Ø	Ø
SCREEN Ø，1	Ø	1
SCREEN 1，Ø	1	Ø
SCREEN 1，1	1	1
CTR＋D	Ø	Ø

※この数字はBG面Øと1を表しています。

説明 BG面を2面持っていますので，表示面とアクティブ（書き込み）面をそれぞれ別々に指定できます。

表示面とアクティブ面が同一の場合は，アクティブ面の指定を省略することができます。

SCREEN Ø，Ø ➡ SCREEN Ø

※CTR キーを押しながらDキーを押すと，表示面とアクティブ面は共にBG面Øになります。

※アクティブ面とは，カーソルが表示されており，カーソルを動かして，実際に表示を変更できる面のことをいいます。

TRON／TROFF

働き トレースモードを実行・解除します。

文法 TRON…トレース実行モード

TROFF…ノーマルモードに戻る

説明 TRON文により，プログラム実行中の行番号を表示させることができます。

TRON文の実行はダイレクトモードでもプログラムモードでもできます。TROFFにすると，TRON命令は終了し，元のモードに戻ります。

```
ｻﾝﾌﾟﾙ ﾌﾟﾛｸﾞﾗﾑ
5  REM *TRON/TROFF
10 FOR N=0 TO 9
20 PRINT"TR=" ;N
30 NEXT
**じっこうけっかはつぎのとおりです**
   #10 #20 TR=0
   #30 #20 TR=1
   #30 #20 TR=2
    :   :   :  :
    :   :   :  :
   #30 #20 TR=9
   #30
```

VCT

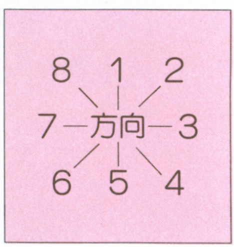
働き DEF MOVEによるアニメキャラクタの移動方向を読み取ります。

省略形 VC.

文法 VCT(n)

n…スプライト番号　Ø～7

VCT（n）の値　　Ø～8



説明 アニメキャラクタが動作しているときに，その方向を番号で与えます。

VCT(n)の値は，DEF SPRITEにおける方向パラメータと同じ値です。ただし，アニメキャラクタが停止している場合や，未定義での値は0です。

※ CRASH の項のサンプルプログラム参照

ファミリーベーシック

FAMILY BASIC™ V3

ふろく



BG TOOLでのキーの働き

● SELECT
COPY
MOVE
CLEAR
FILE
CHAR
を出したりBASICに戻るとき押す

● ESC キーを押した後で押すと，BASICに戻る

● キャラクタメニューを次の8個に替える
（SHIFTを押しながら押すと，前の8個になる）

● 「∧」を左右に動かして選ぶ



● 「□」のある場所を消す

● 「∧」で選んだものを「□」の場所におく

● 「>」で選んだ命令を実行する

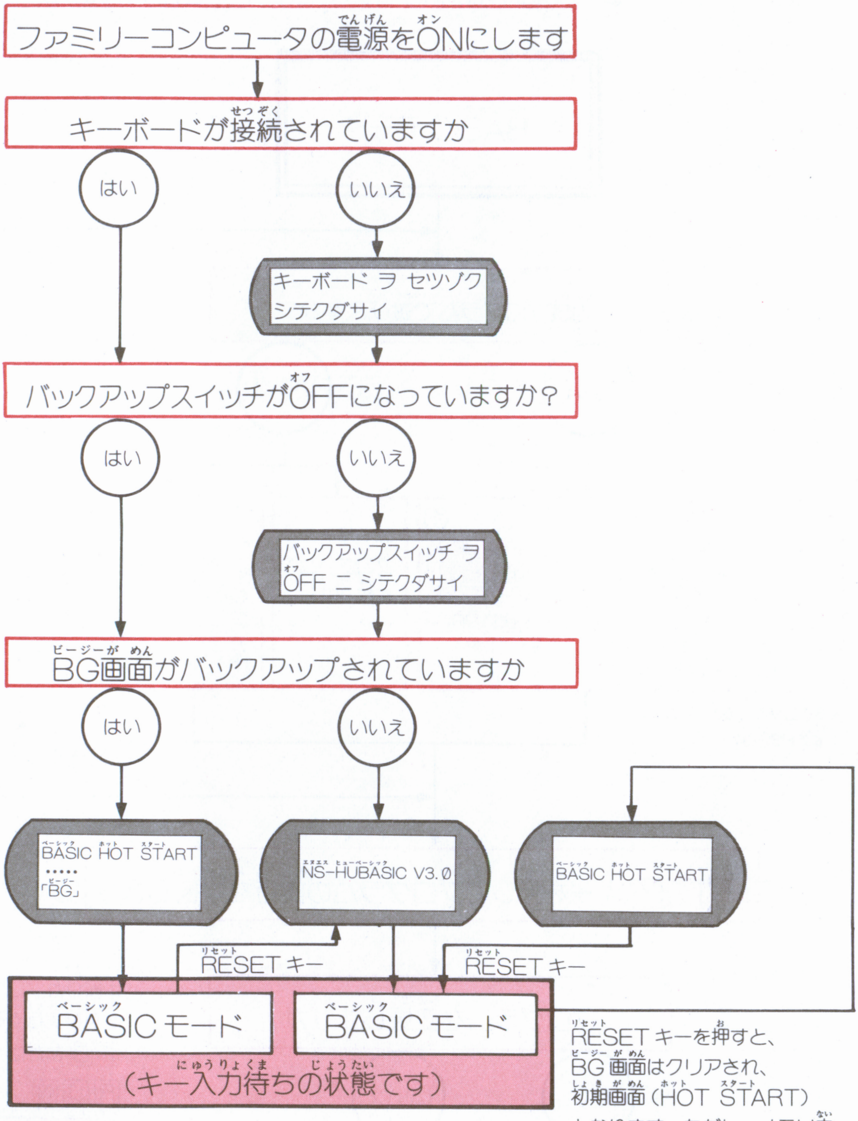
● 色を変える

● 「□」を動かしておく場所を決める

● 「>」を上下に動かして命令を選ぶ

スタート(電源ONから)の手順

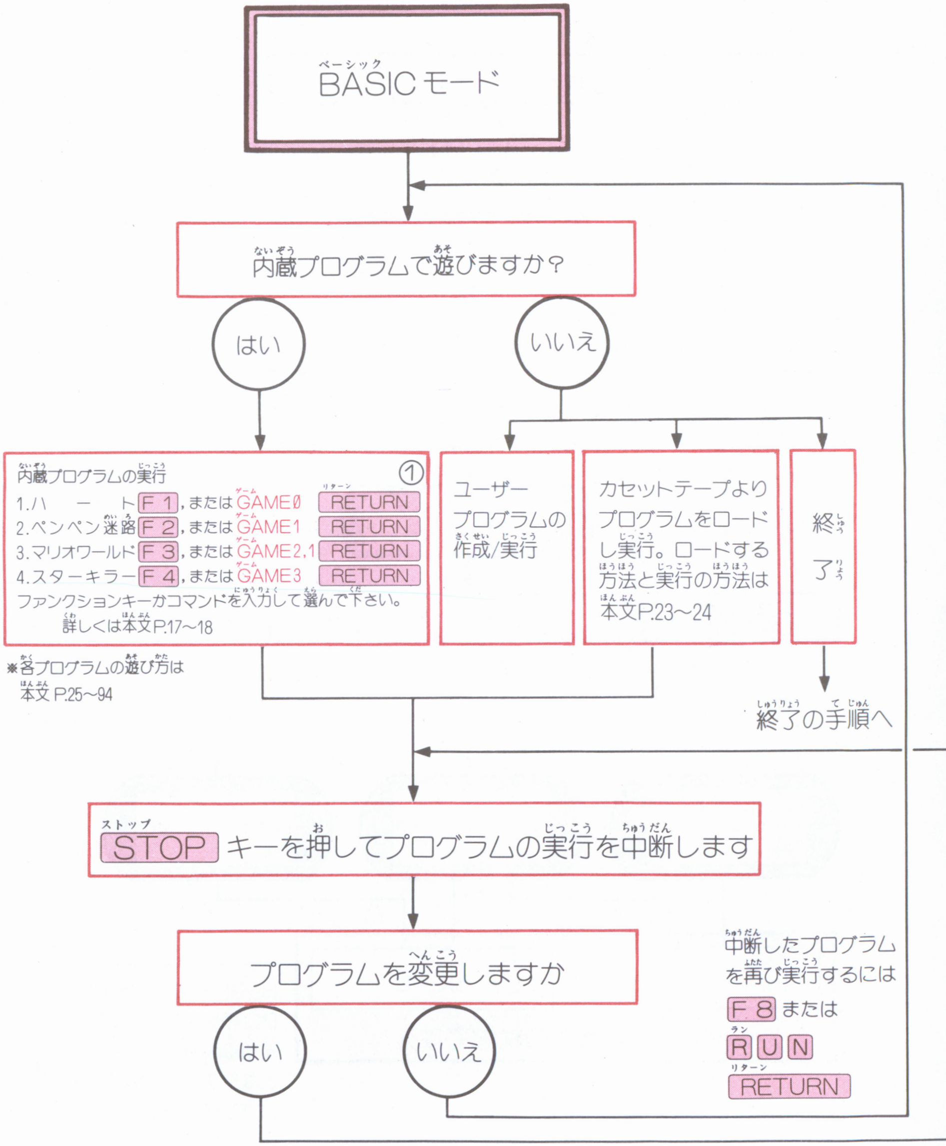
●本文P.14〜16をご覧ください。

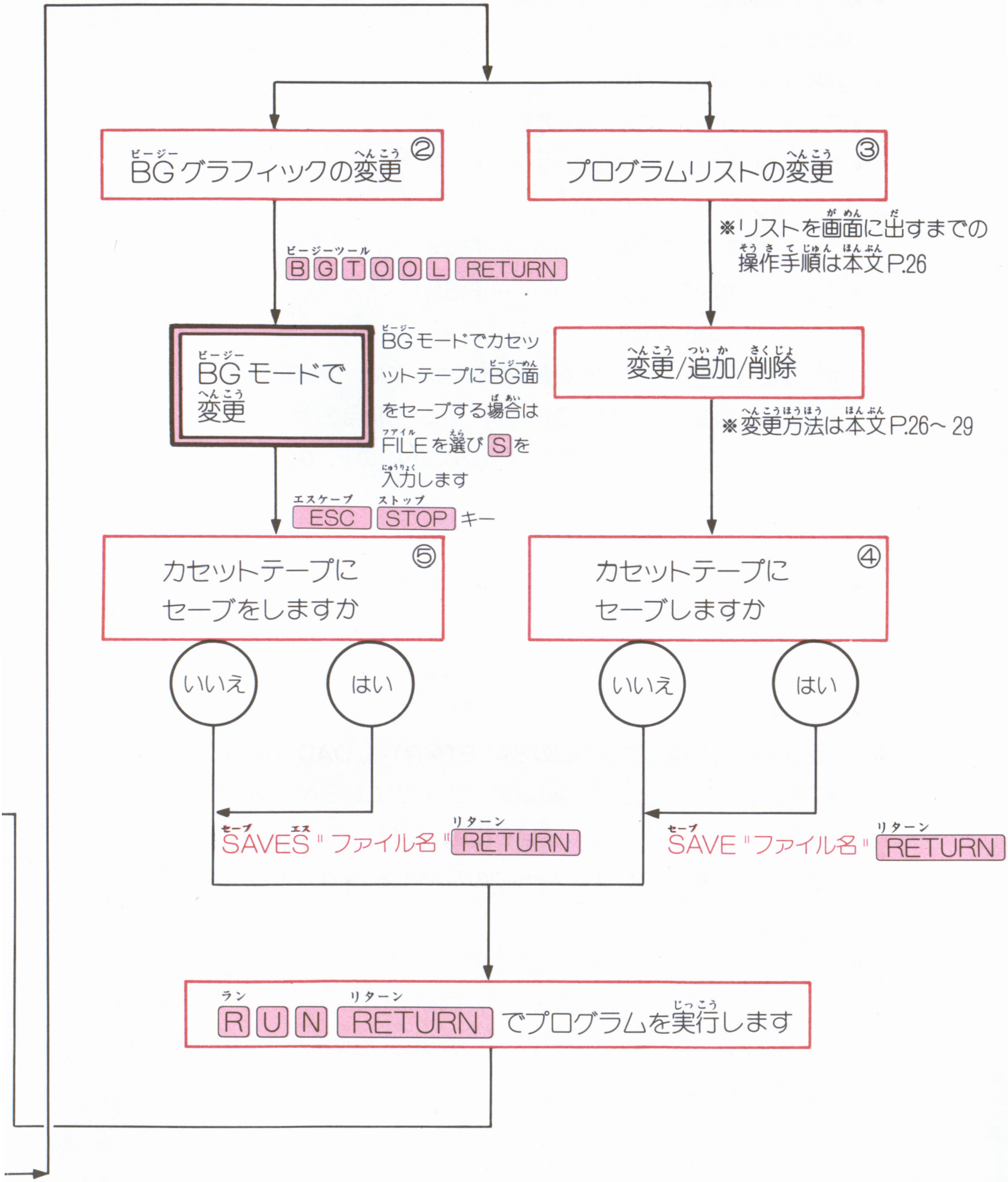


RESET キーを押すと、
BG 画面はクリアされ、
初期画面 (HOT START)
となります。ただし、メモリ内
のプログラムはクリアされません。

内蔵プログラムの実行と変更

●①～⑤の操作についてはP.124をご覧ください。




BGグラフィックの変更 ②
プログラムリストの変更 ③
B G T O O L RETURN
※リストを画面に出すまでの操作手順は本文P.26
BGモードで変更
BGモードでカセットテープにBG面をセーブする場合はFILEを選びSを入力します
変更/追加/削除
※変更方法は本文P.26～29
ESC STOP キー
カセットテープにセーブをしますか ⑤
カセットテープにセーブしますか ④
いいえ
はい
いいえ
はい
SAVES "ファイル名" RETURN
SAVE "ファイル名" RETURN
R U N RETURN でプログラムを実行します

●前ページの図のうち，番号のついた操作については，以下のページを，それぞれご覧ください。

① 内蔵プログラムの選び方

・ファンクションキーで呼び出す方法‥‥P.17
・コマンドを入力して呼び出す方法‥‥‥P.17

② BGグラフィックの変更方法*

・ペンペン迷路のBG変更‥‥‥‥‥‥‥‥P.44
・マリオワールドのBG変更‥‥‥‥‥‥‥P.62
・スターキラーのBG変更‥‥‥‥‥‥‥‥P.84

※BGモードで，キャラクタを選ぶ方法については，ファミリーベーシック取扱説明書の「BG GRAPHIC」(P.40～43）の項目を参照してください。また，BGTOOL 命令については，本ハンドブックのP.100 を参照してください。

③ プログラムリストの変更方法

・ハート‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥P.31
・ペンペン迷路‥‥‥‥‥‥‥‥‥‥‥‥‥‥P.45
・マリオワールド‥‥‥‥‥‥‥‥‥‥‥‥‥P.63
・スターキラー‥‥‥‥‥‥‥‥‥‥‥‥‥‥P.80

④ カセットテープへのプログラムのSAVE(保存)／LOAD（呼び出し）の方法

※ファミリーベーシック取扱説明書の「プログラムのSAVE(保存)，LOAD(呼び出し)」(P.45～48）の項目を参照してください。

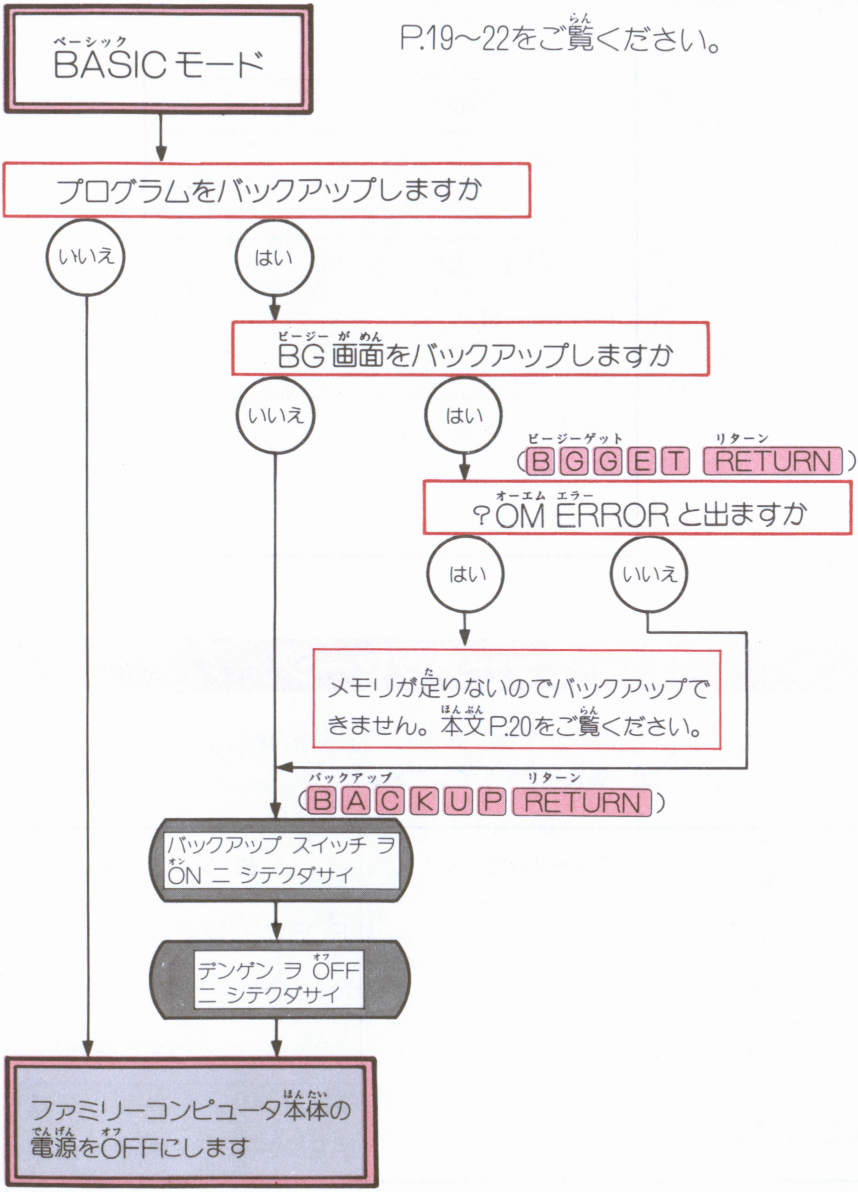
⑤ カセットテープへのBGグラフィックのSAVES(保存)／LOADS(呼び出し)の方法

●SAVES／LOADS 命令による方法 P.23

※BGモードのFILEによる方法については，ファミリーベーシック取扱説明書の「プログラムのSAVE(保存)，LOAD(呼び出し)」(P.45～48）の項目を参照してください。

終了(電源OFF)の手順

●メモリバックアップの方法は、本文P.19～22をご覧ください。



メモリマップ

《V3からの変更点》

アドレス	領域
&H6000〜&H6FFF	ユーザー（RAM）
&H7000〜&H7FFF	未使用域
&H8000〜&HCFFF	BASIC インタ・プリタ
&HD000〜&HD3FF	BG画面（マリオワールド用）
&HD400〜&HDBFD	GAME Ø
&HDBFE〜&HE681	GAME 1
&HE682〜&HF307	GAME 2
&HF308〜&HFFF9	GAME 3
&HFFFA〜&HFFFF	システム用

エラーメッセージ

※ もし，プログラム実行中に，エラーメッセージが表示され，エラーの原因がわからないときは，下記の一覧表を参照してください。

エラーコード	エラー表示	エラーメッセージ	説明
Ø	NF	NEXT without FOR	FORがないのにNEXTがある。
1	SN	Syntax error	文法がまちがっている。
2	RG	RETURN without GOSUB	GOSUBがないのにRETURNがある。
3	OD	Out of DATA	READで読むべきデータがDATA文に用意されていない。

4	IL	Illegal function call	ステートメントや関数の呼び方がまちがっている。
5	OV	Overflow	演算結果が許容範囲を越えた。
6	OM	Out of memory	メモリ不足である。
7	UL	Undefined line Number	GOTO, GOSUB, IFなどで指定した分岐先の行番号がない。
8	SO	Subscript out of range	配列変数の添字が規定外である。
9	DD	Duplicate Definition	配列が2重に定義されている。
10	DZ	Division by zero	Øによる除算をした。
11	TM	Type mismatch	変数の型が一致しない。
12	ST	String too long	文字が31文字を越えている。
13	FT	Formula Too complex	式が複雑すぎる。たとえば，（　）が異常に多い場合。
14	CC	Can't continue	CONTによってプログラムの実行を再開できない。
15	UF	Undefind function	未定義の関数を呼んだ
16	MO	Missing operand	パラメータの必要な命令に指定がない。
17	TP	Tape read ERROR	カセットテープからデータが正しく読めない。
18	NR	No RESUME	ON ERROR GOTOで分岐先にRESUME文がない。
19	RE	RESUME without error	RESUME文のあるルーチンへON ERROR GOTO以外から分岐した。
20	NB	No BG data	BGデータがRAM領域にない。
21	UP	Unprintable error	コードNO.Ø～NO.2Øまでに含まれないエラー。

※コードNO.15，18～21はV3で新たに追加されたものです。

スプライトの関数値とアニメキャラクタの状態の対応表

	アニメキャラクタの状態			
関　数	未定義	定義時	動作中	停　止
MOVE(n)	0	0	−1	0
CRASH(n)	−2	1又は0	−1又は0〜7	−1又は0〜7
VCT(n)	0	0〜8	0〜8	0〜8
XPOS(n)	0	120	(x, y)	(x, y)
YPOS(n)	0	120		

アニメキャラクタや，BGグラフィックで使用するキャラクタの色・形については，これまでのファミリーベーシック「取扱説明書」P.113，裏表紙およびキャラクタマップをご覧ください。

※16種類のアニメキャラクタには，次の数字が割り当てられています。改造をするときこの数字を入れ替えると，アニメキャラクタが変わります。

0：マリオ	6：車	12：レーザー
1：レディ	7：スピナー	13：シェルクリーパー（カメさん）
2：ファイターフライ	8：スターキラー	14：サイドステッパー（カニさん）
3：アキレス	9：スターシップ	15：ニットピッカー（トリさん）
4：ペンペン	10：爆発	
5：ファイアーボール	11：ニタニタ	

DEF MOVE系とDEF SPRITE系の命令対照表

	分類 / 機能	DEF SPRITE系 ユーザーが定義するアニメキャラクタ	DEF MOVE系 特定の動きをするアニメキャラクタ
1	表示を可能にする	SPRITE ON	
2	表示を不可能にする	SPRITE OFF	
3	定義する	DEF SPRITE～	DEF MOVE～
4	未定義の状態にする	RUN 命令での初期化	CAN nØ, n1,～
5	動かす、または表示する	SPRITE n, X, Y	MOVE nØ, n1, ～
6	動きを止める	SPRITE n, X, Y	CUT nØ, n1, ～
7	消す	SPRITE n	ERA nØ, n1, ～
8	初めの位置を決める	SPRITE n, X, Y	POSITION n, X, Y
9	現在の位置を知る	ユーザーのプログラムにより実現できる	XPOS(n), YPOS(n)
10	動いているか止まっているか		MOVE(n)
11	動いているときの方向を知る		VCT(n)
12	重なりの判定		CRASH(n)

BGキャラクタをリストに使う方法

内蔵プログラムのリストを見ると，リスト中に直接BGキャラクタが描き込まれている部分があります。

たとえば「ハート」(GAMEØ)の300行の途中にある，

```
PRINT"●"
```

などです。このようなBGキャラクタは，直接キーで打ち込むことができません。BGキャラクタをリストに使うときには，次のようにしてください。

たとえばハタのBGキャラクタを使った

```
250 PRINT"⚑"
```

という行を例にしましょう。まず，行番号を付けずに

PRINT CHR$(199) RETURN

とキーを押します。これで画面に⚑が出てきます。そうしたら⚑の上にカーソルを動かして，INSキーを10回ほど押してください。これで押した回数だけ，⚑が右に動いていきます。適当なところまで右に動いたら，

250 PRINT"

とキーを押します。このとき⚑が消えてしまう場合は，さらにINSキーを押して，⚑を右に動かしてください。ここまでの操作で，画面が

```
250 PRINT"
```

となっていればいいのですが，たいていは⚑が少し右に行きすぎています。そのときはカーソルを⚑の上に動かして，ちょうど「"」の右にくるまでDELキーを押してください。

最後に⚑のすぐ右に，「"」を付ければでき上がりです。

最初に打ち込んだ

```
PRINT CHR$(199)
```

の199を184～255の間でいろいろ変えると，ほかのBGキャラクタをリストに使うことができます。数字とBGキャラクタの関係は，付録のキャラクタテーブル(P.131)を参照してください。

キャラクタテーブルB

※キャラクタテーブルA、およびA～Dを含むキャラクタテーブルBは、ファミリーベーシック取扱説明書をご覧ください。

	Ø	1	2	3	4	5	6	7
E	184	185	186	187	188	189	19Ø	191
F	192	193	194	195	196	197	199	199
G	2ØØ	2Ø1	2Ø2	2Ø3	2Ø4	2Ø5	2Ø6	207
H	2Ø8	2Ø9	21Ø	211	212	213	214	215
I	216	217	218	219	22Ø	221	222	223
J	224	225	226	227	228	229	23Ø	231
K	232	233	234	235	236	237	238	239
L	24Ø	241	242	243	244	245	246	247
M	248	249	25Ø	251	252	253	254	255

キャラクタコード表

ファミリーベーシックのキャラクタコードを、一覧表で示します。BASICのCHR$(n)、ASC("A$")による文字とコードの変換は、これに対応しています。キャラクタコード表A、またはBのキャラクタ、文字、記号は、バックグラウンド、スプライトの各面で使用できます。(どの面にどちらの表を使用するかの指定はCGEN命令を使用します)

●キャラクタコード表A(主に、スプライト用)

コード(10進数)は、キャラクタテーブルAでアニメキャラクタの四隅に表記してある番号に対応します。

コード(10進数)	コード(16進数)	説明	コード(10進数)	コード(16進数)	説明	コード(10進数)	コード(16進数)	説明	コード(10進数)	コード(16進数)	説明
0	00	マリオ (WALK1)	32	20	レディ (WALK2)	64	40	アキレス (左1)	96	60	ペンペン (左歩1)
1	01		33	21		65	41		97	61	
2	02		34	22		66	42		98	62	
3	03		35	23		67	43		99	63	
4	04	マリオ (WALK2)	36	24	レディ (WALK3)	68	44	アキレス (左2)	100	64	ペンペン (左歩2)
5	05		37	25		69	45		101	65	
6	06		38	26		70	46		102	66	
7	07		39	27		71	47		103	67	
8	08	マリオ (WALK3)	40	28	レディ (JUMP)	72	48	アキレス (左上 1)	104	68	ペンペン (正面)
9	09		41	29		73	49		105	69	
10	0A		42	2A		74	4A		106	6A	
11	0B		43	2B		75	4B		107	6B	
12	0C	マリオ (JUMP)	44	2C	レディ (スリップ 着地)	76	4C	アキレス (左上 2)	108	6C	ペンペン (後)
13	0D		45	2D		77	4D		109	6D	
14	0E		46	2E		78	4E		110	6E	
15	0F		47	2F		79	4F		111	6F	
16	10	マリオ (スリップ 着地)	48	30	レディ (はしご)	80	40	アキレス (上1)	112	70	ファイアーボール(1)
17	11		49	31		81	51		113	71	
18	12		50	32		82	52		114	72	
19	13		51	33		83	53		115	73	
20	14	マリオ (はしご)	52	34	レディ (DOWN)	84	54	アキレス (上2)	116	74	ファイアーボール(2)
21	15		53	35		85	55		117	75	
22	16		54	36		86	56		118	76	
23	17		55	37		87	57		119	77	
24	18	マリオ (DOWN)	56	38	ファイターフライ(1)	88	58	ニタニタ (1)	120	78	車 (左1)
25	19		57	39		89	59		121	79	
26	1A		58	3A		90	5A		122	7A	
27	1B		59	3B		91	5B		123	7B	
28	1C	レディ (WALK1)	60	3C	ファイターフライ(2)	92	5C	ニタニタ (2)	124	7C	車 (左2)
29	1D		61	3D		93	5D		125	7D	
30	1E		62	3E		94	5E		126	7E	
31	1F		63	3F		95	5F		127	7F	

● アニメキャラクタを定義する場合は、このコード表にあるコード(10進数、または16進数)で指定してください。
● 0～255までの10進と16進の換算表としてもご利用ください。

コード(10進数)	コード(16進数)	説明
128	80	車(左上1)
129	81	
130	82	
131	83	
132	84	車(左上2)
133	85	
134	86	
135	87	
136	88	車(上1)
137	89	
138	8A	
139	8B	
140	8C	車(上2)
141	8D	
142	8E	
143	8F	
144	90	スピナー(1)
145	91	
146	92	
147	93	
148	94	スピナー(2)
149	95	
150	96	
151	97	
152	98	スターキラー(左)
153	99	
154	9A	
155	9B	
156	9C	スターキラー(左上)
157	9D	
158	9E	
159	9F	

コード(10進数)	コード(16進数)	説明
160	A0	スターキラー(上)
161	A1	
162	A2	
163	A3	
164	A4	スターシップ(左)
165	A5	
166	A6	
167	A7	
168	A8	スターシップ(左上)
169	A9	
170	AA	
171	AB	
172	AC	スターシップ(上)
173	AD	
174	AE	
175	AF	
176	B0	爆発(1)
177	B1	
178	B2	
179	B3	
180	B4	爆発(2)
181	B5	
182	B6	
183	B7	
184	B8	シェルクリーパー(1)
185	B9	
186	BA	
187	BB	
188	BC	シェルクリーパー(2)
189	BD	
190	BE	
191	BF	

コード(10進数)	コード(16進数)	説明
192	C0	サイドステッパー(1)
193	C1	
194	C2	
195	C3	
196	C4	サイドステッパー(2)
197	C5	
198	C6	
199	C7	
200	C8	ニットピッカー(1)
201	C9	
202	CA	
203	CB	
204	CC	ニットピッカー(2)
205	CD	
206	CE	
207	CF	
208	D0	レーザー(よこ)
209	D1	
210	D2	レーザー(ななめ)
211	D3	
212	D4	レーザー(たて)
213	D5	
214	D6	スプライト用ぬりつぶしの色(1)、(2)、(3)
215	D7	
216	D8	
217	D9	ミュージックボード用カーソル(1)、(2)、(3)
218	DA	
219	DB	
220	DC	第1カーソル
221	DD	第2カーソル
222	DE	ランプ(1)
223	DF	ランプ(2)

コード(10進数)	コード(16進数)	説明
224	E0	はねペン(1)
225	E1	
226	E2	
227	E3	
228	E4	
229	E5	
230	E6	
231	E7	
232	E8	はねペン(2)
233	E9	
234	EA	
235	EB	
236	EC	
237	ED	
238	EE	
239	EF	
240	F0	1
241	F1	2
242	F2	3
243	F3	4
244	F4	COMPUTERの文字
245	F5	
246	F6	
247	F7	
248	F8	
249	F9	
250	FA	OPERATORの文字
251	FB	
252	FC	
253	FD	
254	FE	
255	FF	

●キャラクタコード表B（主に、バックグラウンド画用、及びキーボードの文字、記号、BG GRAPHICに使用できるキャラクタです。）

対応する文字の欄に（　）で書いてある番号は、キャラクタテーブルBのキャラクタに対応します。（113ページ）

コード (10進数)	コード (16進数)	対応する 文字	コード (10進数)	コード (16進数)	対応する 文字	コード (10進数)	コード (16進数)	対応する 文字	コード (10進数)	コード (16進数)	対応する 文字
0	00	(AØ)	32	20		64	40	@	96	60	ア
1	01	(A1)	33	21	!	65	41	A	97	61	イ
2	02	(A2)	34	22	"	66	42	B	98	62	ウ
3	03	(A3)	35	23	#	67	43	C	99	63	エ
4	04	(A4)	36	24	$	68	44	D	100	64	オ
5	05	(A5)	37	25	%	69	45	E	101	65	カ
6	06	(A6)	38	26	&	70	46	F	102	66	キ
7	07	(A7)	39	27	'	71	47	G	103	67	ク
8	08	(BØ)	40	28	(	72	48	H	104	68	ケ
9	09	(B1)	41	29	)	73	49	I	105	69	コ
10	0A	(B2)	42	2A	*	74	4A	J	106	6A	サ
11	0B	(B3)	43	2B	+	75	4B	K	107	6B	シ
12	0C	(B4)	44	2C	,	76	4C	L	108	6C	ス
13	0D	(B5)	45	2D	-	77	4D	M	109	6D	セ
14	0E	(B6)	46	2E	.	78	4E	N	110	6E	ソ
15	0F	(B7)	47	2F	/	79	4F	O	111	6F	タ
16	10	(CØ)	48	30	0	80	50	P	112	70	チ
17	11	(C1)	49	31	1	81	51	Q	113	71	ツ
18	12	(C2)	50	32	2	82	52	R	114	72	テ
19	13	(C3)	51	33	3	83	53	S	115	73	ト
20	14	(C4)	52	34	4	84	54	T	116	74	ナ
21	15	(C5)	53	35	5	85	55	U	117	75	ニ
22	16	(C6)	54	36	6	86	56	V	118	76	ヌ
23	17	(C7)	55	37	7	87	57	W	119	77	ネ
24	18	(DØ)	56	38	8	88	58	X	120	78	ノ
25	19	(D1)	57	39	9	89	59	Y	121	79	ハ
26	1A	(D2)	58	3A	:	90	5A	Z	122	7A	ヒ
27	1B	(D3)	59	3B	;	91	5B	｢	123	7B	フ
28	1C	(D4)	60	3C	<	92	5C	¥	124	7C	ヘ
29	1D	(D5)	61	3D	=	93	5D	｣	125	7D	ホ
30	1E	(D6)	62	3E	>	94	5E	^	126	7E	マ
31	1F	(D7)	63	3F	?	95	5F	-	127	7F	ミ

●バックグラウンド用に直接キャラクタを定義する場合は、このコード表にあるコード（10進数、または16進数）で指定してください。
ただし、コード(16進数)00～1Fは、システムで使用しているため直接指定することはできません。
●Ø～255までの10進と16進の換算表としてもご利用ください。

コード（10進数）	コード（16進数）	対応する文字
128	80	ム
129	81	メ
130	82	モ
131	83	ヤ
132	84	ユ
133	85	ヨ
134	86	ラ
135	87	リ
136	88	ル
137	89	レ
138	8A	ロ
139	8B	ワ
140	8C	ン
141	8D	ヲ
142	8E	ァ*
143	8F	ィ*
144	90	ゥ*
145	91	ェ*
146	92	ォ*
147	93	ャ*
148	94	ュ*
149	95	ョ*
150	96	ッ*
151	97	ガ
152	98	ギ
153	99	グ
154	9A	ゲ
155	9B	ゴ
156	9C	ザ
157	9D	ジ
158	9E	ズ
159	9F	ゼ

コード（10進数）	コード（16進数）	対応する文字
160	A0	ゾ
161	A1	ダ
162	A2	ヂ
163	A3	ヅ
164	A4	デ
165	A5	ド
166	A6	バ
167	A7	ビ
168	A8	ブ
169	A9	ベ
170	AA	ボ
171	AB	パ
172	AC	ピ
173	AD	プ
174	AE	ペ
175	AF	ポ
176	B0	□
177	B1	°
178	B2	[
179	B3	]
180	B4	©
181	B5	×
182	B6	÷
183	B7	ファ
184	B8	(EØ)
185	B9	(E1)
186	BA	(E2)
187	BB	(E3)
188	BC	(E4)
189	BD	(E5)
190	BE	(E6)
191	BF	(E7)

コード（10進数）	コード（16進数）	対応する文字	
192	C0	(FØ)	マリオ用
193	C1	(F1)	
194	C2	(F2)	
195	C3	(F3)	
196	C4	(F4)	
197	C5	(F5)	
198	C6	(F6)	
199	C7	(F7)	
200	C8	(GØ)	LANDSCAPE
201	C9	(G1)	
202	CA	(G2)	
203	CB	(G3)	
204	CC	(G4)	
205	CD	(G5) 星(1)	
206	CE	(G6) 星(2)	
207	CF	(G7) ボール	
208	D0	(HØ)	ドンキーコングJr.用
209	D1	(H1)	
210	D2	(H2)	
211	D3	(H3)	
212	D4	(H4)	
213	D5	(H5)	
214	D6	(H6)	
215	D7	(H7)	
216	D8	(IØ)	島
217	D9	(I1)	
218	DA	(I2)	
219	DB	(I3)	
220	DC	(I4)	
221	DD	(I5)	迷路
222	DE	(I6)	
223	DF	(I7)	

コード（10進数）	コード（16進数）	対応する文字	
224	E0	(JØ)	迷路
225	E1	(J1)	
226	E2	(J2)	
227	E3	(J3)	
228	E4	(J4)	
229	E5	(J5)	
230	E6	(J6)	
231	E7	(J7)	
232	E8	(KØ)	グラフ用ケイ線
233	E9	(K1)	
234	EA	(K2)	
235	EB	(K3)	
236	EC	(K4)	
237	ED	(K5)	
238	EE	(K6)	
239	EF	(K7)	
240	F0	(LØ)	
241	F1	(L1)	
242	F2	(L2)	
243	F3	(L3)	ボード用ワク
244	F4	(L4)	
245	F5	(L5)	
246	F6	(L6)	
247	F7	(L7)	
248	F8	(MØ)	インクツボ
249	F9	(M1)	
250	FA	(M2)	
251	FB	(M3)	
252	FC	(M4)	
253	FD	(M5)(1)	ヌリツブシ
254	FE	(M6)(2)	
255	FF	(M7)(3)	

*は小文字です。

ファミリーベーシックV3の仕様

●ソフト：1．BASIC（BG GRAPHICを含む）
　　　　2．ハート
　　　　3．ペンペン迷路
　　　　4．マリオワールド
　　　　5．スターキラー

項目		仕様
ROM		プログラム用ROM……………8ビット×32K×1個 キャラクタゼネレータ用ROM………8ビット×8K
RAM		ワーキング及びメモリバックアップ用　S－RAM ……8ビット×4K
表示能力	文字表示	28文字×24行（BASIC時）
表示能力	文字構成	8×8ドットマトリックス文字，英文字，数字， カタカナ，英記号，特殊記号，キャラクタ
表示能力	表現色数	52色（白および，黒を含む）
表示能力	動画（スプライト）	16スプライト（但し，1スプライト　16×16ドット または，8×8ドット） 256×24Øドット（BASIC時）
サウンド		2個の方形波音 1個の三角波音 1個の効果音
メモリバックアップ		BASICカセット内に，単三乾電池を内蔵する事により BASICプログラムや，BGグラフィックも一時的に記憶させておく事ができる。

寸法	カセット（HVC-VT）	11Ø（W）×11Ø（D）×17（H）
重量	カセット（HVC-VT）	95g（電池含まず）
使用条件		使用温度Ø～4Ø℃、使用湿度35～75％
附属品		ハンドブック

※仕様および外観の一部を予告なく変更する場合がありますのでご了承ください。

BASICの仕様水準

文字の種類	数字・英文字・カナ・記号
数の表現範囲（整数）	1Ø進（−32768〜+32767） 16進（&HØØØØ〜&HFFFF） 文字列（Ø〜31文字）
変数名の種類	英字で始まる先頭2文字を識別。長さ255桁
行番号の範囲	Ø〜65535
テキスト行の最大桁	255桁
配列の大きさ	2次元まで要素数制限なし（メモリの範囲内）
マルチステートメント	可能、：（コロン）で区切る
サブルーチン、ネスト	制限なし（メモリの範囲内）
FOR NEXTループ	制限なし（メモリの範囲内）
編集機能	スクリーンエディタ
画面モード	BGグラフィック面、スプライト面、バックグラウンド面 バックドロップ面
グラフィック精度	バックグラウンド面（28文字×24行） スプライト面（256×24Øドット） 1キャラクタ（8×8ドット）
カラー	カラーゼネレータ52色発生
フィルター機能	カラーØ〜7の着色可能
サウンド機能	音階、テンポ、3重音、音色
マイク音入力機能	音入力の有無検出
コントローラ入力	I II コントローラ （方向入力、トリガー入力可能）
ファイル機能	カセットテープ（1,2ØØボー）
命令数	基本1ØØ
アニメキャラクタ設定	既設16種類より選択

※数の表現範囲である−32768〜+32767を越える数値を扱う演算結果については保証されません。

取り扱い上のご注意

1. V3カセットを着脱するときには，必ずファミリーコンピュータ本体の電源スイッチを切ってください。

2. V3カセットを差し込むときは，カセットやファミリーコンピュータ本体に，無理な力が加わらないよう充分注意して，最後までしっかりと差し込んでください。

3. V3カセットを抜き取るときは，イジェクトバーを使用し，静かに抜き取ってください。

4. V3カセットを，熱のこもる場所やほこりの多い場所に置かないでください(故障の原因になります)。

5. V3カセットでのご使用中に異常がありましたときには，すぐに使用を中止して，お買い求めの販売店または最寄の任天堂(株)本社・各支店・営業所に，ご相談ください。

6. V3カセットの分解，改造は絶対におやめください。それ以後のカセットおよび本体への動作異常については，保障いたしませんのでご注意ください。

乾電池取り扱い上のご注意

メモリバックアップのためには，単三乾電池2本が必要です。乾電池は，誤った使い方をしますと，液漏れや破裂する恐れがありますので，次の点にご注意ください。

1. 乾電池の⊕と⊖の向きを，V3カセットの表示どおり，正しく入れてください。

2. 新しい乾電池と一度使用した乾電池を，一緒に使わないでください。

3. 種類の違う乾電池を，一緒に使わないでください。同じ形状でも，電圧の異なるものがあります。

4. 乾電池は使えなくなったり，長時間使わないときは，V3カセットから取りはずしておいてください。

ご注意

このハンドブックは、NS−HUBASICに基いて作成されています。

（1）　システムソフトウエアおよび本書の内容は、改良のために予告なく変更することがありますので、バージョンナンバー（NS−HUBASIC　V3．0）には、特にご注意されるよう、お願いいたします。

（2）　本機のシステムソフトウエアならびに本書の内容を、当社に無断で複製することを禁止します。

（3）　本機は、非常に複雑な機能および組み合わせを有する製品であり、出荷に際して、本書を含め充分なチェックをして万全を期しておりますが、万一ご使用中ご不審な点、お気付きのことがありましたら、最寄の任天堂(株)本社、各支店・営業所までご連絡ください。
なお、運用した結果生じる影響については、責任を負いかねますので、あらかじめご了承ください。

※万一、本品に故障が生じました場合、お買い上げの日から6ヵ月以内であれば、お取り換えさせて頂きます。

MEMO

アルファベット順命令語索引

ファミリーベーシックV3™ ハンドブック

1985年2月7日　初版第1刷発行

発行人　山内　溥

発行所　任天堂株式会社

〒605 京都市東山区福稲上高松町60番地
TEL　075(541)6111(代表)

印刷所　株式会社和多田印刷所

©Nintendo 1985

禁無断転載

任天堂株式会社

本　　社　〒605 京都市東山区福稲上高松町６０番地　TEL（075）541－6111番（代表）
東京支店　〒101 東京都千代田区神田須田町１丁目22　TEL（03）254－1781番（代表）
大阪支店　〒542 大阪市南区長堀橋筋１丁目３２　TEL（06）271－5514番（代表）
名古屋営業所　〒451 名古屋市西区幅下２丁目１８番９号　TEL（052）571－2506番（代表）
札幌営業所　〒060 札幌市中央区北九条西18丁目２番地　TEL（011）621－0513番（代表）
岡山営業所　〒700 岡山市奉還町４丁目４番１１号　TEL（0862）52－1821番（代表）

ファミリーベーシック™ V3

任天堂